全国计算机技术与软件专业技术资格（水平）考试参考用书

软件设计师考试应试指导

（第2版）

全国计算机专业技术资格考试办公室推荐

何明 朱宏洁 秦立山 庄琪 主编

清华大学出版社

全国计算机技术与软件专业技术资格(水平)考试参考用书

软件设计师考试应试指导(第 2 版)

何　明　朱宏洁　秦立山　庄　琪　主　编

清华大学出版社
北　京

内 容 简 介

本书以国家最新颁布的软件设计师考试大纲和教程为依据，以对考生进行综合指导、全面提高考生应试能力为原则，在深入研究考试真题基础上结合考前辅导班教师的实际教学经验编写而成。

全书共分 13 章，每章章节开始均设置有“考核说明”板块，简要概括考生需要了解和掌握的内容。书中精选历年真题穿插在知识点的讲解中，有利于考生理解知识点。每章结尾设置有“应试加油站”，该板块分为考频统计和解题技巧两部分，以引导考生掌握重点内容，提高解题能力和综合应用能力。同时每章还有过关习题，方便读者一点一练，巩固提高。书后附有两套模拟试卷并给以详细的分析与解答，可以让考生积累考试经验。

本书主要面向参加软件设计师考试的应试者，同时也可作为高等院校相关课程的辅导书，以及培训班的教材。

图书在版编目(CIP)数据

软件设计师考试应试指导/何明等主编. --2 版. --北京：清华大学出版社，2015
全国计算机技术与软件专业技术资格(水平)考试参考用书
ISBN 978-7-302-40467-5

Ⅰ. ①软…　Ⅱ. ①何…　Ⅲ. ①软件设计—工程技术人员—资格考核—自学参考资料　Ⅳ. ①TP311.5

中国版本图书馆 CIP 数据核字(2015)第 121614 号

责任编辑：魏　莹
封面设计：常学影
版式设计：杨玉兰
责任校对：马素伟
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**32.75　　**字　数：**797 千字
版　　次：2012 年 5 月第 1 版　2015 年 7 月第 2 版　　**印　次：**2015 年 7 月第 1 次印刷
印　　数：1～3000
定　　价：59.00 元

产品编号：061430-01

前　言

在信息技术和软件产业快速发展的推动下，计算机技术与软件专业技术资格(水平)考试(通常简称“软考”)已经成为我国国家级的 IT 专业人员从业资格考试。软考通过后颁发的资格证书不仅在全国范围内有效，部分级别的考试还可以与中日、中韩相应级别的考试互认。因此软考的权威性已得到社会和广大用人单位的认可。

软考不仅注重广度，也有一定的深度，因此难度大，考生平均通过率较低。为了更好地服务于考生，引导考生在较短的时间内掌握解题要领，并顺利通过考试，我们本着“标准、实用、严谨”的原则，组织多位一线教师和全国计算机技术与软件专业技术资格(水平)考试试题研究组的成员精心分析历年真题和考试大纲，去冗存精，推出这样一本切实为广大考生服务的软件设计师考试用书。全书突出以下特点。

(1) 紧扣国家最新考试大纲和教程，对大纲规定的知识点进行了细化和深化，精讲重点、难点。

(2) 结合教程、真题讲解和模拟试题三者为一体，达到“一本通”的效果，可为考生节省复习时间和花费。

(3) 精选历年真题穿插在知识点的讲解中，有利于考生理解知识点。真题是备考的最佳资料，是考生把握考试动态的最好途径，本书正文对最近 3 年考试的真题进行了分类解析。

(4) 每章均设有“应试加油站”，包括“考频统计”和“解题技巧”两个子板块，通过统计最近 3 年考试涉及的真题，突出考试重点，方便考生有所侧重；“解题技巧”精选已考真题，附有详尽解析，可以帮助考生掌握解题的各种技巧，熟练解题方法。

(5) 书后附有两套全真模拟试题，便于考生提前进行实战演练。试卷的命题风格、考点分布和难度水平与真题一致。

全书共 13 章，由何明、朱宏洁、秦立山、庄琪主编，参与本书资料收集和编写的还有何光明、王珊珊、毛幸甜、卢振侠、周海霞、吴婷、石雅琴、郑爱琴、曹冬梅、陈珍、陈凤、杨橙、陈莉萍等。本书在编写过程中，编者参考了许多相关的书籍和资料，在此对这些参考文献的作者表示感谢。

因编者水平有限，书中难免存在错漏和不妥之处，欢迎广大读者批评指正。

编　者

目　　录

第1章

计算机系统知识

本章要点

- 中央处理单元的组成
- 数据表示方法
- 校验码
- 计算机体系结构
- 指令系统
- 存储系统
- 输入输出技术
- 总线结构
- 计算机可靠度分析
- 信息安全和网络威胁
- 数据加密技术
- 认证技术

1.1 计算机系统的基础知识

考核说明：本节主要介绍中央处理单元、数据表示和校验码。CPU 的组成、指令控制、各种寄存器的作用，数据表示中的补码、浮点数是考核重点，要理解透彻。校验码较难，在最近几次考试中没有出现。

1.1.1 计算机系统硬件

计算机硬件系统的基本组成如图 1-1 所示。中央处理单元(CPU)是硬件系统的核心，存储器是计算机系统中的记忆设备。

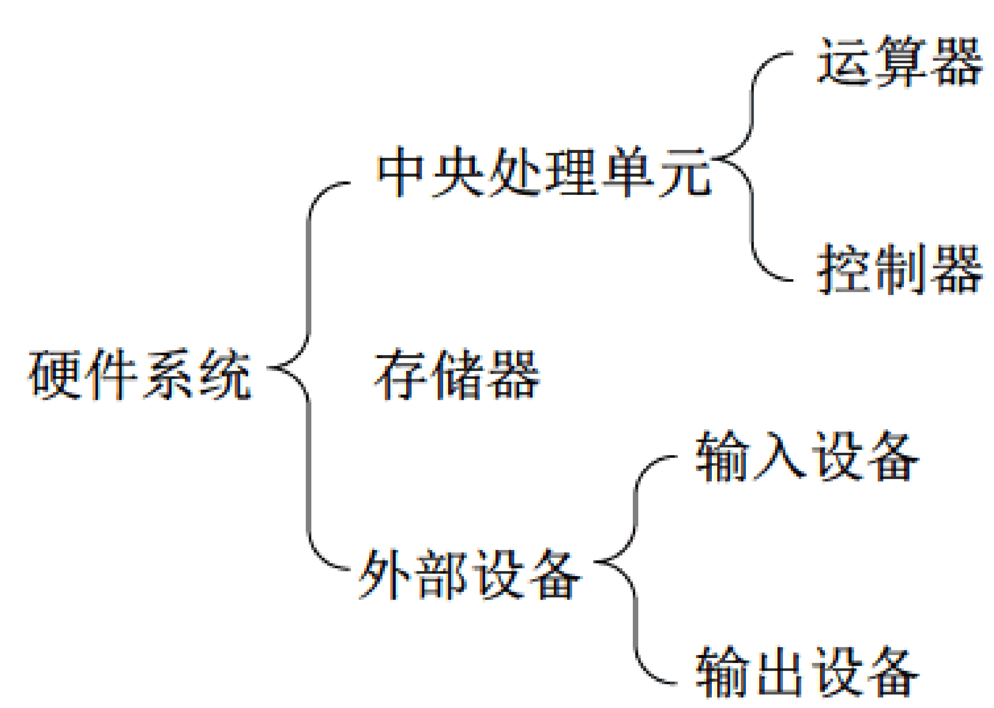


图 1-1 计算机硬件系统的基本组成

1.1.2 中央处理单元

1. CPU 的功能

CPU 的功能有程序控制、操作控制、时间控制和数据处理。对数据的加工处理是 CPU 最根本的任务。

2. CPU 的组成

CPU 主要由运算器、控制器、寄存器组和内部总线等部件组成。

1) 运算器

运算器是数据加工处理部件，用于完成各种算术和逻辑运算。它由算术逻辑单元(Arithmetic and Logic Unit，ALU)、累加寄存器(AC)、数据缓冲寄存器(DR)和状态条件寄存器(PSW)组成。

2) 控制器

运算器只能完成运算，而控制器用于控制整个 CPU 的工作，它决定了计算机运行过程的自动化。控制器一般包括指令控制逻辑、时序控制逻辑、总线控制逻辑和中断控制逻辑等几个部分。其中指令控制逻辑是考试重点。

指令控制过程分为取指令、指令译码、按指令操作码执行、形成下一条指令地址等

步骤。

(1)　程序计数器(PC)。

PC 用来存放当前要执行的指令地址。在程序开始执行前，将程序的起始地址送入 PC，这时可以根据 PC 的内容从内存中取出第一条指令执行，CPU 将自动修改 PC 的内容，以使其保持的总是将要执行的下一条指令的地址。如果为单字长指令，则 PC+1，若为双字长指令，则 PC+2，依次类推；当遇到转移指令时，后继指令的地址根据当前指令的地址加上一个向前或向后转移的位移量得到，或者根据转移指令给出的直接转移的地址得到。

(2)　指令寄存器(IR)。

IR 用来保存正在执行的指令。通常 IR 中的指令在整个指令执行期间保持不变，由它来控制当前指令正在执行的操作。

(3)　指令译码器(ID)。

ID 用来对 IR 中的指令进行译码。指令分为操作码和地址码两部分，每一条指令的基本操作由操作码指定。ID 对指令中的操作码进行分析和解释，识别该指令规定的操作，向操作控制器发出具体的控制信号，控制各部件工作，从而完成所需的功能。

(4)　地址寄存器(AR)。

AR 保存当前 CPU 所访问的内存单元的地址。由于内存和 CPU 存在操作速度上的差异，因此需要使用 AR 保持地址信息，直到内存的读/写操作完成为止。

真题链接

【例 1-1】 属于 CPU 中算术逻辑单元的部件是__(3)__。(2014 年 11 月真题 3)

(3) A. 程序计数器　　B. 加法器

C. 指令寄存器　　D. 指令译码器

【解析】 算术逻辑单元(Arithmetic and Logic Unit，ALU)主要完成对二进制数据的定点算术运算(加、减、乘、除)、逻辑运算(与、或、非、异或)以及移位操作。

【答案】 B

【例 1-2】 在 CPU 中，常用来为 ALU 执行算术逻辑运算提供数据并暂存运算结果的寄存器是__(1)__。(2014 年 5 月真题 1)

(1) A. 程序计数器　　B. 状态寄存器

C. 通用寄存器　　D. 累加寄存器

【解析】 在运算器中，累加寄存器是一个专门存放算术或逻辑运算的操作数和运算结果的寄存器，能进行加、减、读出、移位、循环移位和求补等操作，是运算器的主要部分。

【答案】 D

3)　寄存器组

寄存器组可分为专用寄存器和通用寄存器。运算器和控制器中的寄存器是专用寄存器，其作用是固定的。通用寄存器的用途广泛并可由程序员规定其用途，其数目因处理器的不同而有所差异。

3. 多核 CPU

多核 CPU 是指在一个单芯片上集成两个甚至更多个处理器内核，其中每个内核都有自

己的逻辑单元、控制单元、中断处理器和运算单元，一级 Cache、二级 Cache 共享或独有，其部件的完整性和单核处理器内核相比完全一致。

1.1.3 数据表示

各种数据在计算机中表示的形式称为机器数，其特点是数的符号用 0、1 表示。机器数又分为无符号数和带符号数两种。无符号数表示正数，在机器数中没有符号位。对于带符号数，机器数的最高位是表示正、负的符号位，其余二进制位表示数值。带符号的机器数可采用原码、反码、补码、移码等编码方法。

1. 原码、反码、补码和移码

1) 原码

在原码表示中，机器数的最高一位是符号位，0 代表正号，1 代表负号，余下各位是数的绝对值。数值 0 的原码表示方法有两种形式：$[+0]_{原}=00000000$，$[-0]_{原}=10000000$。

2) 反码

在反码表示中，机器数的最高一位是符号位，0 代表正号，1 代表负号。正数的反码与原码相同；负数的反码则是其绝对值按位求反。数值 0 的反码表示方法有两种形式：$[+0]_{反}=00000000$，$[-0]_{反}=11111111$。

3) 补码

在补码表示中，机器数的最高一位是符号位，0 代表正号，1 代表负号。正数的补码与原码相同；负数的补码等于其反码的末尾加 1。数值 0 的补码有唯一的表示形式，即$[+0]_{补}=[-0]_{补}=00000000$。

4) 移码

移码表示法是在数 X 上增加一个偏移量来定义的，常用于表示浮点数中的阶码。对于定点整数 X，$[X]_{移}=2^{n-1}+X$。如果知道了一个数的补码，则将补码的最高位取反，即得到该数的移码。

2. 定点数和浮点数

1) 定点数

所谓定点数，就是小数点的位置固定不变的数。小数点的位置通常有两种约定方式：定点整数(纯整数，小数点在最低有效数值位之后)和定点小数(纯小数，小数点在最高有效数值位之前)。

2) 浮点数

浮点数是小数点位置不固定的数，它能表示更大范围的数。一个二进制数 N 可以表示为更为一般的形式：$N=2^E\times F$，其中 E 称为阶码，F 称为尾数。用阶码和尾数表示的数叫作浮点数，这种表示数的方法称为浮点表示法。

在浮点表示法中，阶码通常为带符号的纯整数，尾数为带符号的纯小数。浮点数的表示格式如图 1-2 所示。其中，尾数决定精度，阶码决定表示范围。最适合表示浮点数阶码的数字编码是移码。

阶符	阶码	数符	尾数

图 1-2　浮点数的表示格式

为了充分利用尾数来表示更多的有效数字，通常采用规格化浮点数。规格化就是将尾数的绝对值限定在区间[0.5,1]。

3)　工业标准 IEEE 754

IEEE 754 是由 IEEE 制定的有关浮点数的工业标准，被广泛采用。该标准的表示形式如下：

$$(-1)^S 2^E (b_0 b_1 b_2 b_3 \cdots b_{p-1})$$

式中：$(-1)^S$ 为该浮点数的数符，当 S 为 0 时表示正数，S 为 1 时表示负数；E 为指数(阶码)，用移码表示；$(b_0 b_1 b_2 b_3 \cdots b_{p-1})$ 为尾数，其长度为 P 位，用原码表示。

对于单精度浮点数，P=23，S=1，E=8，指数偏移量为+127。在 IEEE 754 标准中，约定小数点左边隐含有一位，通常这位数就是 1，因此单精度浮点数尾数的有效尾数为 24 位，即尾数为 1.**…*。

4)　浮点数的运算

两个浮点数进行加减运算时，要经过对阶、求尾数和(差)、结果规格化并判溢出、舍入处理和溢出判别等步骤。

浮点数相乘，其积的阶码等于两乘数的阶码相加，积的尾数等于两乘数的尾数相乘。浮点数相除，其商的阶码等于被除数的阶码减去除数的阶码，商的尾数等于被除数的尾数除以除数的尾数。乘除运算的结果都需要进行规格化处理并判断阶码是否溢出。

真题链接

【例 1-3】某机器字长为 n，最高位是符号位，其定点整数的最大值为 (2) 。(2014 年 5 月真题 2)

(2)　A. 2^n-1　　B. $2^{n-1}-1$　　C. 2^n　　D. 2^{n-1}

【解析】由于最高位是符号位，因此最大的定点整数是：

$$0\underbrace{11111\cdots\cdots111}_{n-1\text{ 个 }1}$$

最高位0表示正数，值为$2^0+2^1+2^2+\cdots+2^{n-2}=2^{n-1}-1$。

【答案】B

1.1.4　校验码

计算机系统运行时，各个部件之间要进行数据交换，有两种方法可以确保数据在传送过程中正确无误，一是提高硬件电路的可靠性，二是提高代码的校验能力，包括查错和纠错。通常使用校验码的方法来检测传送的数据是否出错。所谓码距，是指一个编码系统中任意两个合法编码之间至少有多少个二进制位不同。

1. 奇偶校验

奇偶校验是一种简单有效的校验方法。其基本思想是，通过在编码中增加一位校验位来使编码中 1 的个数为奇数(奇校验)或者为偶数(偶校验)，从而使码距变为 2。对于奇校验，它可以检测代码中奇数位出错的编码，但不能发现偶数位出错的情况。

常用的奇偶校验码有三种：水平奇偶校验码、垂直奇偶校验码和水平垂直校验码。

2. 海明码

海明码的构成方法是：在数据位之间插入 k 个校验码，通过扩大码距来实现检错和纠错。设数据位是 n 位，校验位是 k 位，则 n 和 k 必须满足 $2^k-1 \geqslant n+k$ 的关系。

3. 循环冗余校验码

循环冗余校验码(CRC)是一种循环码，其特征是信息字段和校验字段的长度可以任意选定，在局域网中有广泛应用。

生成 CRC 码的基本原理是：任意一个由二进制位串组成的代码都可以和一个系数仅为 0 和 1 取值的多项式一一对应。例如：代码 1010111 对应的多项式为 $x^6+x^4+x^2+x+1$。

CRC 码集选择的原则是：若设码字长度为 N 位，信息字段为 K 位，校验字段为 R 位 ($N=K+R$)，则对于 CRC 码集中的任一码字，存在且仅存在一个 R 次多项式 $g(x)$，使得

$$V(x)=A(x)g(x)=x^R m(x)+r(x)$$

其中，$m(x)$为 K 次信息多项式，$r(x)$为 R−1 次校验多项式。

$$g(x)=g_0+g_1x+g_2x^2+\cdots+g_Rx^R$$

通常将 $g(x)$称为生成多项式，即所有合法的码字都可以由 $g(x)$生成。数据通信的发送方通过指定的 $g(x)$产生 CRC 码字，接收方则通过该 $g(x)$来验证收到的 CRC 码字。

在求 CRC 编码时，采用的是模 2 运算。模 2 运算中加减运算的规则是按位运算，不发生借位和进位。

真题链接

【例 1-4】海明码利用奇偶性检错和纠错，通过在 n 个数据位之间插入 k 个检验位，扩大数据编码的码距。若 $n=48$，则 k 应为 (3) 。(2014 年 5 月真题 3)

(3) A. 4　　B. 5　　C. 6　　D. 7

【解析】n 和 k 满足关系：$2^k-1 \geqslant n+k$，题中 n=48，k 为 6 可满足上述关系式。

【答案】C

1.2　计算机体系结构

考核说明：本节主要介绍计算机体系结构的分类和指令系统，其中指令系统部分较为重要，经常会考到，要掌握寻址方式、CISC 和 RISC 的特点、流水指令控制方式，会计算吞吐率。

1.2.1　计算机体系结构的分类

计算机体系结构是指计算机的概念性结构和功能属性。

从宏观上按处理机的数量进行分类，可以分为单处理系统、并行处理与多处理系统、分布式处理系统。

在微观上按并行程度分类，有 Flynn 分类法、冯泽云分类法、Handler 分类法和 Kuck 分类法。

(1) M. J. Flynn 把计算机系统结构分为四大类：单指令流单数据流(SISD)、单指令流多数据流(SIMD)、多指令流单数据流(MISD)、多指令流多数据流(MIMD)。

(2) 冯泽云将计算机系统结构分为四种：字串行位串行(WSBS)、字并行位串行(WPBS)、字串行位并行(WSBP)、字并行位并行(WPBP)。

(3) 汉德勒(Wolfgang Handler)把计算机的硬件结构分为三个层次：处理机级、每个处理机中的算逻单元级、每个算逻单元中的逻辑门电路级。

(4) 库克(David J. Kuck)把系统结构分为四类：单指令流单执行流(SISE)、单指令流多执行流(SIME)、多指令流单执行流(MISE)、多指令流多执行流(MIME)。

真题链接

【例 1-5】Flynn 分类法基于信息流特征将计算机分成 4 类，其中 (6) 只有理论意义而无实例。(2014 年 11 月真题 6)

(6) A. SISD　　B. MISD
C. SIMD　　D. MIMD

【解析】按照 Flynn 分类法，根据计算机中指令和数据的并行状况可把计算机分成：

单指令流单数据流(SISD)——传统的计算机包含单个 CPU，它从存储在内存中的程序那里获得指令，并作用于单一的数据流。

单指令流多数据流(SIMD)——单个的指令流作用于一个以上的数据流上。例如有数据 4、5 和 3、2，一个单指令执行两个独立的加法运算：4+5 和 3+2，就被称为单指令流多数据流。SIMD 的一个例子就是一个数组或向量处理系统，它可以对不同的数据并行执行相同的操作。

多指令流单数据流(MISD)——用多个指令作用于单个数据流的情况实际上很少见。这种冗余多用于容错系统。

多指令流多数据流(MIMD)——这种系统类似于多个 SISD 系统。实际上，MIMD 系统的一个常见例子是多处理器计算机，如 Sun 的企业级服务器。

【答案】B

1.2.2　指令系统

指令系统指的是一个 CPU 所能够处理的全部指令的集合，是一个 CPU 的根本属性。一条指令一般包括两个部分：操作码和地址码。操作码用于指明操作的类型，地址码主要用于指明操作数及运算结果存放的地址。

1. 寻址方式

如何对指令中的地址字段进行解释，以获得操作数的方法或获得程序转移地址的方法称为寻址方式。常见的寻址方式有如下几种。

- 立即寻址：操作数作为指令的一部分而直接写在指令中，这种操作数称为立即数。
- 寄存器寻址：指令所要的操作数已存储在某寄存器中，或把目标操作数存入寄存器。
- 直接寻址：指令所要的操作数存放在内存中，在指令中直接给出该操作数的有效地址。
- 寄存器间接寻址：操作数在存储器中，操作数的有效地址用 SI、DI、BX 和 BP 四个寄存器之一来指定。
- 寄存器相对寻址：操作数在存储器中，其有效地址是一个基址寄存器(BX、BP)或变址寄存器(SI、DI)的内容和指令中的 8 位/16 位偏移量之和。
- 基址加变址寻址方式：操作数在存储器中，其有效地址是一个基址寄存器(BX、BP)和一个变址寄存器(SI、DI)的内容之和。
- 相对基址加变址寻址：操作数在存储器中，其有效地址是一个基址寄存器(BX、BP)的值、一个变址寄存器(SI、DI)的值和指令中的 8 位/16 位偏移量之和。

2. CISC 和 RISC

CISC(Complex Instruction Set Computer，复杂指令集计算机)的基本思想是：进一步增强原有指令的功能，用更为复杂的新指令取代原先由软件子程序完成的功能，实现软件功能的硬化，导致机器的指令系统越来越庞大而复杂。

RISC(Reduced Instruction Set Computer，精简指令集计算机)的基本思想是：通过减少指令总数和简化指令功能，降低硬件设计的复杂度，使指令能单周期执行，并通过优化编译，提高指令的执行速度，采用硬布线控制逻辑，优化编译程序。

3. 指令控制方式

指令控制方式有顺序方式、重叠方式和流水方式三种。

(1) 顺序方式。顺序方式是指各条机器指令之间顺序串行地执行，执行完一条指令后才取下一条指令，而且每条机器指令内部的各个微操作也是顺序串行地执行。

(2) 重叠方式。重叠方式是指在解释第 x 条指令的操作完成之前，就可开始解释第 $x+i$ 条指令。通常采用的是一次重叠，即在任何时候，指令分析部件和指令执行部件都只有相邻两条指令在重叠解释。

(3) 流水方式。流水方式是把并行性或并发性嵌入到计算机系统中的一种形式，它把重复的顺序处理过程分解为若干子过程，每个子过程能在专用的独立模块上有效地并发工作，如图 1-3 所示。

在概念上，“流水”可以看成 “重叠”的延伸。差别仅在于“一次重叠”只是把一条指令解释分解为两个子过程，而“流水”则是分解为更多的子过程。

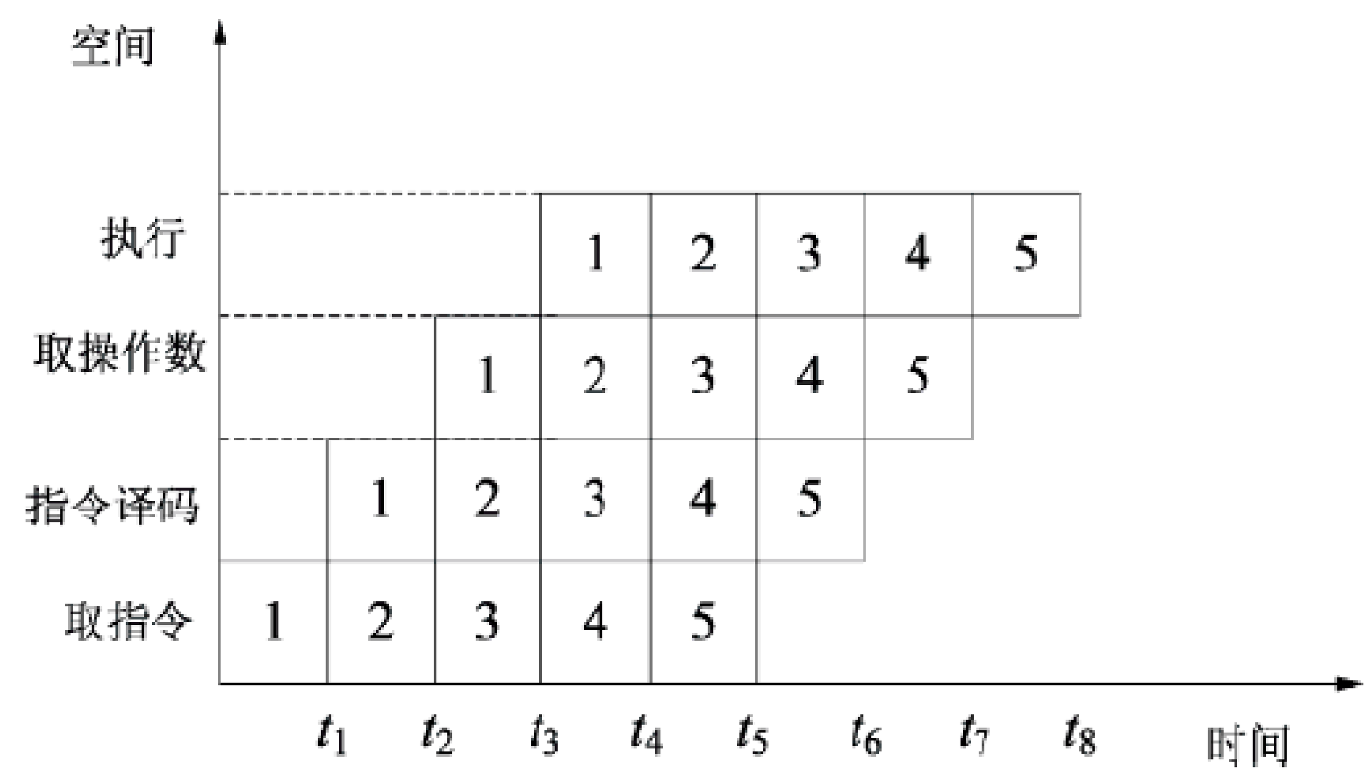


图 1-3　流水处理的时空图

4. 吞吐率和流水建立时间

吞吐率是指单位时间内流水线处理机流出的结果数。对指令而言，就是单位时间内执行的指令数。如果流水线的子过程所用时间不一样，则吞吐率 p 应为最长子过程的倒数，即

$$p = 1/\max\{\Delta t_1, \Delta t_2, \cdots, \Delta t_m\}$$

流水线开始工作，须经过一定时间才能达到最大吞吐率，这就是建立时间。若 m 个子过程所用时间一样，均为Δt_0，则建立时间 $T_0=m\Delta t_0$。

真题链接

【例 1-6】以下关于 RISC 和 CISC 的叙述中，不正确的是__(5)__。(2014 年 11 月真题 5)

(5) A．RISC 通常比 CISC 的指令系统更复杂

B．RISC 通常会比 CISC 配置更多的寄存器

C．RISC 编译器的子程序库通常要比 CISC 编译器的子程序库大得多

D．RISC 比 CISC 更加适合 VLSI 工艺的规整性要求

【解析】(1) 指令系统：RISC 设计者把主要精力放在那些经常使用的指令上，尽量使它们具有简单高效的特色。对不常用的功能，常通过组合指令来完成。而 CISC 计算机的指令系统比较丰富，有专用指令来完成特定的功能。因此，处理特殊任务效率较高。

(2) 存储器操作：RISC 对存储器操作有限制，使控制简单化；而 CISC 机器的存储器操作指令多，操作直接。

(3) 程序：RISC 汇编语言程序一般需要较大的内存空间，实现特殊功能时程序复杂，不易设计；而 CISC 汇编语言程序编程相对简单，科学计算及复杂操作的程序设计相对容易，效率较高。

(4) 设计周期：RISC 微处理器结构简单，布局紧凑，设计周期短，且易于采用最新技术；CISC 微处理器结构复杂，设计周期长。

(5) 应用范围：由于 RISC 指令系统的确定与特定的应用领域有关，故 RISC 机器更适合于专用机；而 CISC 机器则更适合于通用机。

【答案】A

【例 1-7】通常可以将计算机系统中执行一条指令的过程分为取指令、分析和执行指令 3 步。若取指令时间为 $4\Delta t$，分析时间为 $2\Delta t$，执行时间为 $3\Delta t$，则按顺序方式从头到尾执行完 600 条指令所需时间为__(4)__Δt；若按照执行第 i 条，分析第 i+1 条，读取第 i+2 条重叠的流水线方式执行指令，则从头到尾执

行完 600 条指令所需时间为__(5)__Δt。(2014 年 5 月真题 4～5)

(4) A. 2400　　B. 3000　　C. 3600　　D. 5400

(5) A. 2400　　B. 2405　　C. 3000　　D. 3009

【解析】按顺序方式需要执行完一条执行之后再执行下一条指令，执行 1 条执行所需的时间为 $4\Delta t+2\Delta t+3\Delta t=9\Delta t$，执行 600 条指令所需的时间为 $9\Delta t\times600=5400\Delta t$。

若采用流水线方式，则处理过程如图 1-4 所示，可见执行完 600 条执行所需要的时间为 $4\Delta t\times600+2\Delta t+3\Delta t=2405\Delta t$。

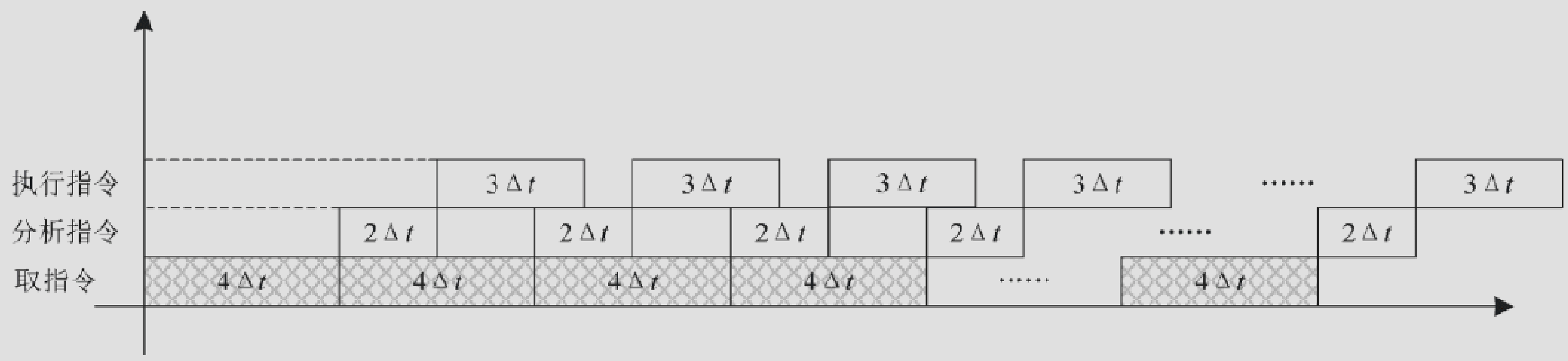


图 1-4　流水线方式的处理过程

【答案】D　B

1.3　存 储 系 统

考核说明：本节主要介绍存储器的层次结构、分类，Cache 存储器，相关存储器，虚拟存储器，外存储器等内容。相对来说，Cache 存储器、外存储器较为重要，需要理解掌握。

1.3.1　存储器的层次结构与分类

1. 层次结构

计算机的三层存储体系结构如图 1-5 所示。

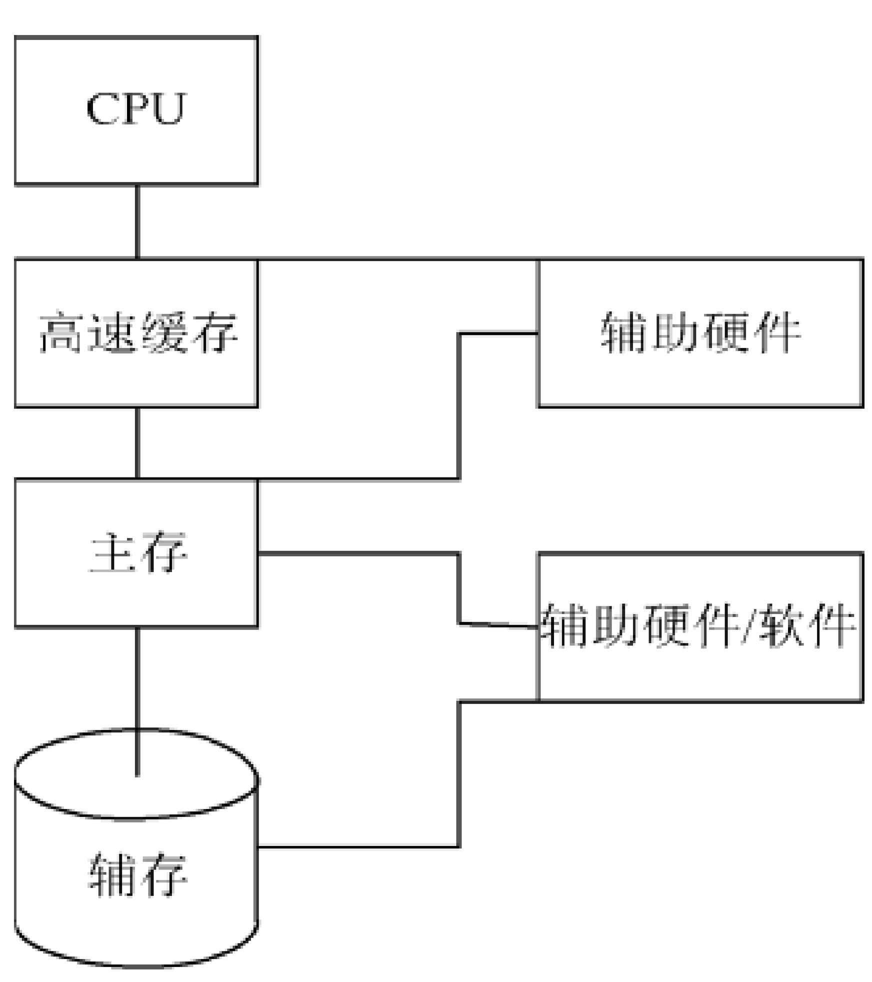


图 1-5　存储器的层次结构

三层存储结构是高速缓存(Cache)、主存储器(MM)和辅助存储器(外存储器)。若将 CPU 的内部寄存器也看作存储器的一个层次，那么存储器的层次分为四层。若有些计算机没有高速缓存，那么存储器的层次分为两层，即只有主存和辅存。

2. 分类

存储器按位置分类，可分为内存和外存。

- 内存(主存)：用来存储当前参与运行的程序和数据，速度快，容量小。
- 外存(辅存)：用来存储目前不参与运行的数据，容量大但速度慢。

存储器按寻址方式分类，可分为随机存储器、顺序存储器和直接存储器。

- 随机存储器(Random Access Memory，RAM)：这种存储器可对任何存储单元存入或读取数据，访问任何一个存储单元所需时间都是相同的。
- 顺序存储器(Sequentially Addressed Memory，SAM)：访问数据所需时间与数据所在存储位置有关。磁带是典型的顺序存储器。
- 直接存储器(Direct Addressed Memory，DAM)：采用介于随机存取和顺序存取之间的一种寻址方式。磁盘是一种直接存取控制器，它对磁道的寻址是随机的，而在一个磁道内，则是顺序寻址。

真题链接

【例 1-8】计算机采用分级存储体系的主要目的是为了解决 (2) 的问题。(2014 年 11 月真题 2)

(2) A. 主存容量不足　　　　B. 存储器读写可靠性

C. 外设访问效率　　　　D. 存储容量、成本和速度之间的矛盾

【解析】为了解决对存储器要求容量大、速度快、成本低三者之间的矛盾，目前通常采用多级存储器体系结构，即使用高速缓冲存储器、主存储器和外存储器。高速缓冲存储器：高速存取指令和数据，存取速度快，但存储容量小。主存储器：存放计算机运行期间的大量程序和数据，存取速度较快，存储容量不大。外存储器：存放系统程序和大型数据文件及数据库，存储容量大，成本低。

【答案】D

【例 1-9】内存按字节编址从 A5000H 到 DCFFFH 的区域，其存储容量为 (4) 。(2014 年 11 月真题 4)

(3) A. 123KB　　B. 180KB　　C. 223KB　　D. 224KB

【解析】DCFFFH – A5000H + 1H = 38000H = 224KB

【答案】D

1.3.2　高速缓存

高速缓存(Cache)是位于 CPU 和主存之间的高速存储子系统。采用高速缓存的主要目的是提高存储器的平均访问速度，使存储器的速度与 CPU 的速度相匹配。Cache 的存在对程序员是透明的，其地址变换和数据块的替换算法均由硬件实现。通常 Cache 被集成到 CPU 内，以提高访问速度，其主要特点是容量小、速度快、成本高。

1. Cache 的组成

Cache 由两部分组成：控制部分和 Cache 部分。Cache 部分用来存放主存的部分复制信息。控制部分的功能是：判断 CPU 要访问的信息是否在 Cache 中，若在即为命中，若不在则没有命中。命中时直接对 Cache 寻址；未命中时，要按照替换原则，决定主存的一块信息放到 Cache 的哪一块里面。

2. Cache 中的地址映像方法

因为处理机访问都是按主存地址访问的，而要从 Cache 中读写信息，这就需要地址映像，即把主存中的地址映射成 Cache 中的地址。地址映像的方法有三种：直接映像、全相联映像和组相联映像。

3. 替换算法

选择替换算法的目标是使 Cache 获得最高的命中率。常用的替换算法有如下几种。

- 随机替换(RAND)算法：用随机数发生器产生一个要替换的块号，将该块替换出去。
- 先进先出(FIFO)算法：将最先进入的 Cache 信息块替换出去。
- 近期最少使用(LRU)算法：将 Cache 中近期最少使用的信息块替换出去。
- 优化替换(OPT)算法：先执行一次程序，统计 Cache 的替换情况。有了这样的先验信息，在第二次执行该程序时便可以用最有效的方式来替换，达到最优的目的。

4. Cache 的性能分析

若 H 为 Cache 的命中率，t_c 为 Cache 的存取时间，t_m 为主存的访问时间，则 Cache 的等效访问时间 t_a 为

$$t_a = Ht_c + (1-H)t_m$$

使用 Cache 的 CPU 比不使用 Cache 的 CPU 访问存储器的速度提高的倍数 r 可以用下式求得：

$$r = t_m / t_a$$

真题链接

【例 1-10】在程序执行过程中，Cache 与主存的地址映像由 (1) 。(2013 年 11 月真题 1)

(1) A．硬件自动完成　　B．程序员调度

C．操作系统管理　　D．程序员与操作系统协同完成

【解析】Cache 与主存之间的地址映像是由硬件自动完成的，以达到快速访问的目的，这一过程对程序员来说是透明的。

【答案】A

1.3.3 相联存储器与虚拟存储器

1. 相联存储器

相联存储器是一种按内容访问的存储器。其工作原理是把数据或数据的某一部分作为

关键字，将该关键字与存储器中的每一单元进行比较，找出存储器中所有与关键字相同的数据字。

高速缓冲存储器可用在相联存储器中，在虚拟存储器中用来做段表、页表或块表存储器，还可以用在数据库和知识库中。

2. 虚拟存储器

虚拟存储器是由主存、辅存、存储管理单元及操作系统中存储管理软件组成的存储系统。程序员使用该存储系统时，可以使用的内存空间可以远远大于主存的物理空间，但实际上并不存在那么大的主存，故称其为虚拟存储器。虚拟存储器的空间大小取决于计算机的访存能力而不是实际外存的大小，实际存储空间可以小于虚拟地址空间。从程序员的角度看，外存被看作逻辑存储空间，访问的地址是一个逻辑地址(虚地址)，虚拟存储器使存储系统既具有相当于外存的容量又有接近于主存的访问速度。

虚拟存储器的访问也涉及虚地址与实地址的映像、替换算法等，这与 Cache 中的类似。前面我们讲的地址映像以块为单位，而在虚拟存储器中，地址映像以页为单位。设计虚拟存储系统需考虑的指标是主存空间利用率和主存的命中率。

1.3.4　外存储器

外存储器用来存放暂时不用的程序和数据，并且以文件的形式存储。CPU 不能直接访问外存中的程序和数据，只有将其以文件为单位调入主存后才可访问。外存由磁表面存储器(如磁盘、磁带)和光盘存储器构成。

1. 磁盘存储器

磁盘存储器由盘片、驱动器、控制器和接口组成。盘片用来存储信息；驱动器用于驱动磁头沿盘面径向运动以寻找目标磁道位置，驱动盘片以额定速率稳定旋转，并且控制数据的写入和读出；控制器接收主机发来的命令，将它转换成磁盘驱动器的控制命令，并实现主机和驱动器之间数据格式的转换及数据传送，以控制驱动器的读写操作；接口是主机和磁盘存储器之间的连接逻辑。

磁盘容量有两种指标：一种是非格式化容量，它是指一个磁盘所能存储的总位数；另一种是格式化容量，它是指各扇区中数据区容量的总和。计算公式分别如下：

非格式化容量=面数×(磁道数/面)×内圆周长×最大位密度

格式化容量=面数×(磁道数/面)×(扇区数/道)×(字节数/扇区)

2. 光盘存储器

1)　光盘存储器的类型

根据性能和用途，可分为只读型光盘、只写一次型光盘和可擦除型光盘。

2)　光盘存储器的组成及特点

光盘存储器由光学、电学和机械部件等组成。其特点是：记录密度高；存储容量大；采用非接触式读写信息；信息可长期保存；采用多通道记录时数据传输率可超过 200Mbps；

制造成本低；对机械结构的精度要求不高；存取时间较长。

1.3.5 磁盘阵列技术

磁盘阵列是由多台磁盘存储器组成的一个快速、大容量且高可靠的外存子系统。现在常见的廉价冗余磁盘阵列(Redundant Array of Inexpensive Disks，RAID)，就是一种由多块廉价磁盘构成的冗余阵列。虽然 RAID 包含多块磁盘，但是在操作系统下是作为一个独立的大型存储设备出现的。RAID 技术分为几种不同的等级，分别可以提供不同的速度、安全性和性价比，如表 1-1 所示。

表 1-1 廉价冗余磁盘阵列

RAID 级	说 明
RAID-0	RAID-0 是一种不具备容错能力的阵列
RAID-1	RAID-1 是采用镜像容错技术改善可靠性的一种磁盘阵列
RAID-2	RAID-2 是采用海明码进行错误检测的一种磁盘阵列
RAID-3	RAID-3 减少了用于检验的磁盘存储器的台数，从而提高了磁盘阵列的有效容量。一般只有一个检验盘
RAID-4	RAID-4 是一种可独立地对组内各磁盘进行读写的磁盘阵列，该阵列也只用一个检验盘
RAID-5	RAID-5 是对 RAID-4 的一种改进，它不设置专门的检验盘。同一台磁盘上既记录数据，也记录检验信息。这就解决了前面多台磁盘机争用一台检验盘的问题
RAID-6	RAID-6 磁盘阵列采用两级数据冗余和新的数据编码以解决数据恢复问题，在两个磁盘出现故障时仍然能够正常工作。在进行写操作时，RAID-6 分别进行两个独立的校验运算，形成两个独立的冗余数据，并写入两个不同的磁盘

1.4 输入/输出技术

1.4.1 常见的内存与接口的编址方式

1. 内存与接口地址独立的编址方法

内存地址与接口地址完全独立且相互隔离，在使用中内存用于存放程序和数据，而接口就用于寻址外设，在编程序和读程序时很容易使用和辨认。此编址方法的缺点就是用于接口的指令太少，功能太弱。

2. 内存与接口地址统一的编址方法

内存地址与接口地址统一在一个公共的地址空间中，在这些地址空间中拿一些地址分配给接口使用，剩下的就可以归内存使用。优点是原则上用于内存的指令全部都可以用于

接口；缺点在于整个地址空间被分成两部分，一部分分配给接口使用，另一部分分配给内存使用，这经常会导致内存地址不连续。再就是用于内存的指令和用于接口的指令是完全一样的，这在读程序时就要根据参数定义表仔细加以辨认。

1.4.2　CPU 与外设之间的数据传送方式

CPU 与外设之间的数据传送方式有如下几种。

1. 直接程序控制

直接程序控制方式是指在完成数据的输入/输出中，整个数据的输出过程是在 CPU 执行程序的控制下完成的。这种方式还可以分为如下两种。

- 无条件传送方式：无条件地与 CPU 交换数据。
- 程序查询方式：先通过 CPU 查询外设状态，准备好之后再与 CPU 交换数据。

2. 中断控制

中断控制方式利用了中断机制，当 I/O 系统与外设交换数据时，CPU 无须等待，也不必查询 I/O 状态即可以抽身出来处理其他任务，因此提高了系统效率。

1)　中断处理方法

中断处理方法有多中断信号线法、中断软件查询法、菊花链法、总线仲裁法及中断向量表法。

2)　中断优先级控制

在进行优先级控制时解决以下两种情况。

- 当不同优先级的多个中断源同时提出中断请求时，CPU 应优先响应优先级最高的中断源。
- 当 CPU 正在对某一个中断源服务时，若又有比它优先级更高的中断源提出中断请求，CPU 应能暂时中断正在执行的中断服务程序而转去为优先级更高的中断源服务，服务结束后再回到原先被中断的优先级较低的中断服务程序继续执行。

3. 直接存取

直接存取方式是在存储器与 I/O 设备之间直接传送数据，即在内存与 I/O 设备之间传送一个数据块的过程中，不需要 CPU 的任何干预，是一种完全由 DMA 硬件完成 I/O 操作的方式。

4. 输入/输出处理机

输入/输出处理机(IOP)是一个专用处理机，用于完成主机的输入/输出操作。IOP 根据主机的 I/O 命令，完成对外设数据的输入/输出。它的数据传送方法有三种：字节多路方式、选择传送方式和数组多路方式。

真题链接

【例 1-11】中断向量可提供__(2)__。(2013 年 5 月真题 2)

(2) A. I/O 设备的端口地址　　B. 所传送数据的起始地址
C. 中断服务程序的入口地址　　D.主程序的断点地址

【解析】计算机发生中断时，用中断向量指向各个寄存器和程序计数器所指向的内容，或者其中所存的内容，并将中断向量暂时存储在另一个地方，而当执行完别的程序时，可以从暂存的地方将中断向量取出放入原来的位置，从而可以执行原来中断的程序，即中断向量可保存中断服务程序的入口地址。

【答案】C

1.5 总 线 结 构

广义地讲，任何连接两个以上电子元器件的导线都可以称为总线。总线通常分为如下三类。

(1) 内部总线。用于芯片一级的互连。

(2) 系统总线。用于构成计算机各组成部分(CPU、内存和接口等)的连接。

(3) 外部总线。又称通信总线，用于设备一级的互连。

系统总线是微处理器芯片对外引线信号的延伸或映射，是微处理器与片外存储器及 I/O 接口传输信息的通路。系统总线有时也称内总线。目前比较流行的内总线有以下几种。

- ISA(Industry Standard Architecture，工业标准体系结构)总线。
- EISA(Extended Industry Standard Architecture，扩展工业标准体系结构)总线。
- PCI(Peripheral Component Interconnection，外围部件互连)总线。

外部总线的标准有七八十种之多，常见的外总线标准有如下几种。

- RS-232-C 总线。
- SCSI(Small Computer System Interface，小型计算机系统接口)总线。
- USB(Universal Serial Bus，通用串行总线)总线。
- IEEE-1394 总线。

真题链接

【例 1-12】三总线结构的计算机总线系统由 (1) 组成。(2014 年 11 月真题 1)

(1) A. CPU 总线、内存总线和 IO 总线　　B. 数据总线、地址总线和控制总线
C. 系统总线、内部总线和外部总线　　D. 串行总线、并行总线和 PCI 总线

【解析】总线线路都可以被归为三类，即数据总线、地址总线和控制总线。

【答案】B

1.6 计算机安全

考核说明：本节的考核重点是数据的表示和校验码。

1.6.1　计算机安全概述

计算机安全是指计算机资源安全，即要保证计算机资源不受自然和人为的有害因素的威胁和危害。

1. 计算机的安全等级

计算机系统中的三类安全性是指技术安全性、管理安全性及政策法律安全性。重要的安全评估准则有如下几个。

- 美国国防部(DOD)和国家标准局(现更名为 NIST)的可信计算机系统评估准则。
- 欧洲共同体(现为欧洲联盟)的信息技术安全评估准则(ITSEC)。
- ISO/IEC 国际标准。
- 美国联邦标准。其中美国国防部和国家标准局的可信计算机系统评测标准将系统划分为 4 组 7 个等级。

2. 安全威胁

安全威胁是指某个人、物、事件对某一资源的机密性、完整性、可用性或合法性所造成的危害。某种攻击就是威胁的具体实现。安全威胁分为两类：故意(如黑客渗透)或偶然(如信息发往错误的地址)。

网络安全威胁分为以下几种。

(1) 物理威胁。物理威胁是指计算机硬件和存储介质遭受到偷窃、废物搜寻及歼敌活动的威胁。

(2) 网络攻击。计算机网络的使用对数据造成了新的安全威胁。攻击者可通过网络上存在的电子窃听、入侵拨号入网、冒名顶替等方式进行入侵攻击、偷窃和篡改。

(3) 身份鉴别。由于身份鉴别通常是用设置口令的手段实现的，入侵者可通过口令圈套、密码破译等方式扰乱身份鉴别。

(4) 编程威胁。编程威胁是指通过病毒进行攻击的一种方法。

(5) 系统漏洞。系统漏洞也称代码漏洞，通常是由操作系统设计者有意设置的，目的是使用户在失去对系统的访问权时，仍有机会进入系统。入侵者可使用扫描器发现系统漏洞，从而进行攻击。

1.6.2　网络攻击手段

黑客(Hacker)常用的攻击手段主要有口令入侵、放置特洛伊木马、DoS 攻击、端口扫描、网络监听、欺骗攻击、电子邮件攻击等。

1. 口令入侵

所谓口令入侵是指使用某些合法用户的账号和口令登录到目的主机，然后再实施攻击活动。使用这种方法的前提是必须先得到该主机上的某个合法用户的账号，然后再进行合法用户的口令的破译。

2. 放置特洛伊木马

在计算机领域，有一类特殊的程序，黑客通过它来远程控制别人的计算机，把这类程序称为特洛伊木马程序(一般可简称为木马)。特洛伊木马程序一般分为服务器端(Server)和客户端(Client)两部分，服务器端是攻击者传到目标机器上的部分(用来在目标机上监听，等待客户端连接过来)；客户端是用来控制目标机器的部分，放在攻击者的机器上。

3. DoS 攻击

DoS 即拒绝服务，其攻击目的是使计算机或网络无法提供正常的服务。最常见的 DoS 攻击有计算机网络带宽攻击和连通性攻击。

分布式拒绝服务(DDoS)攻击是指借助于客户机/服务器技术，将多个计算机联合起来作为攻击平台，对一个或多个目标发送 DoS 攻击，从而成倍地提高拒绝服务攻击的威力。

4. 端口扫描

端口扫描就是利用 Socket 编程与目标主机的某些端口建立 TCP 连接、进行传输协议的验证等，从而获知目标主机的扫描端口是否处于激活状态、主机提供了哪些服务、提供的服务中是否含有某些缺陷等。常用的扫描方式有 TCP connect()扫描、TCP SYN 扫描、TCP FIN 扫描、IP 段扫描和 FTP 返回攻击等。

5. 网络监听

网络监听是主机的一种工作模式，在这种模式下，主机可以接收到本网段在同一条物理通道上传输的所有信息，而不管这些信息的发送方和接收方是谁。

Sniffer 是一个著名的监听工具，可以监听到网上传输的所有信息。Sniffer 可以是硬件也可以是软件，主要用来接收在网络上传输的信息。

6. 欺骗攻击

欺骗攻击是攻击者创造一个易于误解的上下文环境，以诱使受攻击者进入并且做出缺乏安全考虑的决策。常见的欺骗攻击有 Web 欺骗、ARP 欺骗、IP 欺骗。

7. 电子邮件攻击

电子邮件攻击主要表现为向目标信箱发送电子邮件炸弹。所谓的邮件炸弹实质上就是发送地址不详且容量庞大的邮件垃圾。由于邮件信箱是有限的，当庞大的邮件垃圾到达信箱时，就会把信箱挤爆。

真题链接

【例 1-13】以下关于拒绝服务攻击的叙述中，不正确的是 (8) 。(2014 年 11 月真题 8)

(8) A. 拒绝服务攻击的目的是使计算机或者网络无法提供正常的服务

B. 拒绝服务攻击是不断地向计算机发起请求来实现的

C. 拒绝服务攻击会造成用户密码的泄露

D. DDoS 是一种拒绝服务攻击形式

【解析】拒绝服务攻击是指攻击者想办法让目标机器停止提供服务，是黑客常用的攻击手段之一。其

实对网络带宽进行的消耗性攻击只是拒绝服务攻击的一小部分，只要能够对目标造成麻烦，使某些服务被暂停甚至主机死机，都属于拒绝服务攻击。拒绝服务攻击问题一直得不到合理的解决，因为这是由于网络协议本身的安全缺陷造成的，因此拒绝服务攻击也成了攻击者的终极手法。攻击者进行拒绝服务攻击，实际上是让服务器实现两种效果：一是迫使服务器的缓冲区满，不接收新的请求；二是使用 IP 欺骗，迫使服务器把合法用户的连接复位，影响合法用户的连接。

【答案】 C

1.6.3　病毒防护

1. 计算机病毒

计算机病毒是指编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据，影响计算机使用，并能自我复制的一组计算机指令或者程序代码。

计算机病毒的分类方法有多种，根据其感染的途径以及采用的技术区分，计算机病毒可分为文件型病毒、引导型病毒、宏病毒和目录(链接)型病毒。

1)　文件型病毒

文件型病毒以感染.COM、.EXE 等可执行文件为主，病毒以这些可执行文件为载体，当运行可执行文件时就可以激活病毒。

2)　引导型病毒

引导型病毒不会感染文件，也就是说如果某个磁盘感染了引导型病毒，只要不用它去引导计算机，其中的数据文件将不会受到影响。

3)　宏病毒

宏病毒是一种寄存在文档或模板的宏中的计算机病毒。一旦打开这样的文档，其中的宏就会被执行，于是宏病毒就会被激活，转移到计算机上，并驻留在 Normal 模板上。从此以后，所有自动保存的文档都会“感染”上这种宏病毒，而且如果其他用户打开了感染病毒的文档，宏病毒又会转移到其他计算机上。

4)　目录(链接)型病毒

目录(链接)型病毒会修改文件存储位置信息以达到传染的目的。用户如果利用工具(SCANDISK 或 CHKDSK)检测磁盘，发现大量的文件链接地址错误，说明计算机感染了目录(链接)型病毒。不要试图用上述软件去修复，否则情况会更糟。

2. 蠕虫与木马

蠕虫与木马都是恶意代码，它们会在不被察觉的情况下寄宿到另一段程序中，当这段程序被运行时，就会对计算机和网络系统进行恶意的破坏。

1)　蠕虫

蠕虫是一种通过网络传播的恶性病毒，它具有病毒的一些共性，如传播性、隐蔽性、破坏性等。与普通病毒相比，蠕虫不使用驻留文件即可在系统之间进行自我复制，它的传染目标是互联网内的所有计算机。同时，蠕虫能控制计算机传输文件或信息的功能。

2) 木马

“木马”程序是目前比较流行的病毒文件，但是它不会自我繁殖，也不“刻意”地去感染其他文件，它通过伪装自身以吸引用户下载执行，从而向施种木马者提供打开被种者电脑的门户，使施种者可以任意毁坏、窃取被种者的文件，甚至远程操控被种者的电脑。

一个完整的“木马”程序包含两部分：“服务器”和“控制器”。植入被种者电脑的是“服务器”部分，而所谓的“黑客”正是利用“控制器”进入运行了“服务器”的电脑。

真题链接

【例 1-14】<u>(9)</u> 不是蠕虫病毒。(2014 年 11 月真题 9)

(9) A. 熊猫烧香　　B. 红色代码　　C. 冰河　　D. 爱虫病毒

【解析】“熊猫烧香”是一种经过多次变种的计算机蠕虫病毒，2006 年 10 月 16 日由 25 岁的中国湖北武汉新洲区人李俊编写，2007 年 1 月初肆虐中国大陆网络，它主要通过网络下载的文件植入计算机系统。

“红色代码”病毒是一种新型网络病毒，其传播所使用的技术可以充分体现网络时代网络安全与病毒的巧妙结合，将网络蠕虫、计算机病毒、木马程序合为一体，开创了网络病毒传播的新路，可称为划时代的病毒。

“冰河”木马病毒是国人编写的一种黑客性质的病毒，黑客可以通过网络远程控制感染该病毒的电脑。

“爱虫”病毒，是一种蠕虫病毒，这个病毒可以改写本地及网络硬盘上的某些文件。用户机器染毒以后，邮件系统将会变慢，并可能导致整个网络系统崩溃。

【答案】C

【例 1-15】以下关于木马程序的叙述中，正确的是<u>(7)</u>。(2014 年 5 月真题 7)

(7) A. 木马程序主要通过移动磁盘传播

B. 木马程序的客户端运行在攻击者的机器上

C. 木马程序的目的是使计算机或网络无法提供正常的服务

D. Sniffer 是典型的木马程序

【解析】木马(Trojan)，是指通过特定的程序(木马程序)来控制另一台计算机。木马通常有两个可执行程序：一个是客户端，另一个是服务端。植入对方电脑的是服务端，而黑客则利用客户端进入运行了服务端的电脑。运行了木马程序的服务端以后，会产生一个有着容易迷惑用户的名称的进程，暗中打开端口，向指定地点发送数据(如网络游戏的密码，即时通信软件密码和用户上网密码等)，黑客甚至可以利用这些打开的端口进入电脑系统。

Sniffer，中文可以翻译为嗅探器，是一种基于被动侦听原理的网络分析方式。使用这种技术方式，可以监视网络的状态、数据流动情况以及网络上传输的信息。Sniffer 不是木马程序。

【答案】B

1.6.4 防火墙、漏洞扫描与入侵检测

1. 防火墙的类型

防火墙通过监测、限制、更改跨越防火墙的数据流，尽可能地对外部屏蔽网络内部的信息、结构和运行状况，以此来实现网络的安全保护。

防火墙的基本类型包括以下几种。

- 包过滤型防火墙：通过访问控制表，检查数据流中每个数据包的源地址、目的地址、所用的端口号、协议状态等因素，来确定是否允许该数据包通过。
- 应用网关防火墙：它工作在开放系统互连参考模型(OSI/RM)的应用层，能针对特别的网络应用协议制定数据过滤规则。
- 代理服务器防火墙：它工作在 OSI 模型的应用层，主要使用代理技术来阻断内部网络和外部网络之间的通信，达到隐藏内部网络的目的。
- 状态检测防火墙：也叫自适应防火墙或动态包过滤防火墙。这种防火墙能通过状态检测技术动态记录、维护各个连接的协议状态，并且在网络层和 IP 之间插入一个检查模块，对 IP 包的信息进行分析检测，以决定是否允许通过防火墙。
- 自适应代理防火墙：根据用户的安全策略，动态适应传输中的分组流量。它整合了动态包过滤防火墙技术和应用代理技术，本质上是状态检测防火墙。

2．防火墙的物理特性

防火墙通常至少具有 3 个接口，但许多早期的防火墙只具有 2 个接口。当使用具有 3 个接口的防火墙时，至少会产生 3 个网络，它们的描述如下。

- 内部区域通常就是指企业内部网络或者企业内部网络的一部分。它是互联网络(interconnection network)的信任区域，即受到了防火墙的保护。
- 外部区域通常指 Internet 或者非企业内部网络。它是互联网络中不被信任的区域，当外部区域想要访问内部区域的主机和服务时，通过设置防火墙就可以实现有限制的访问。
- 非军事区(DMZ，也称隔离区)是一个隔离的网络，或几个网络。位于非军事区中的主机或服务器被称为堡垒主机。一般在非军事区内可以放置 Web 服务器和 Mail 服务器等。非军事区对于外部用户通常是可以访问的，这种方式允许外部用户访问企业的公开信息，但却不允许他们访问企业内部网络。

3. 漏洞扫描系统

漏洞扫描系统是一种自动检测远程或本地主机安全性弱点的程序。通过使用漏洞扫描系统，系统管理员能够发现所维护的 Web 服务器的各种 TCP 端口的分配、提供的服务、Web 服务软件版本和这些服务及软件呈现在 Internet 上的安全漏洞。

漏洞形成的原因形形色色，最常见的漏洞主要包含以下类型：CGI 脚本、POP3、FTP、SSH、HTTP、SMTP、IMAP、后门、RPC、DNS 漏洞等。

4. 入侵检测

入侵检测是一种主动保护自己免受攻击的网络安全技术，即通过对计算机网络或计算机系统中的若干个关键点收集信息并对其进行分析，从中发现网络或系统中是否有违反安全策略的行为和被攻击的迹象。入侵检测的软件与硬件的组合便是入侵检测系统(IDS)。

真题链接

【例 1-16】网络系统中，通常把__(7)__置于 DMZ 区。(2014 年 11 月真题 7)

(7) A．网络管理服务器　　B．Web 服务器

C．入侵检测服务器　　D．财务管理服务器

【解析】DMZ 是为了解决安装防火墙后外部网络不能访问内部网络服务器的问题，而设立的一个非安全系统与安全系统之间的缓冲区，这个缓冲区位于企业内部网络和外部网络之间的小网络区域内，在这个小网络区域内可以放置一些必须公开的服务器设施，如企业 Web 服务器、FTP 服务器和论坛等。

【答案】B

【例 1-17】以下关于包过滤防火墙和代理服务防火墙的叙述中，正确的是__(9)__。(2014 年 5 月真题 9)

(9) A. 包过滤成本技术实现成本较高，所以安全性能高

B. 包过滤技术对应用和用户是透明的

C. 代理服务技术安全性较高，可以提高网络整体性能

D. 代理服务技术只能配置成用户认证后才建立连接

【解析】包过滤技术是一种基于网络层、传输层的安全技术，优点是简单实用，实现成本较低；同时，包过滤操作对于应用层来说是透明的，它不要求客户与服务器程序做任何修改。但包过滤技术无法识别基于应用层的恶意入侵，如恶意的 Java 小程序以及电子邮件中附带的病毒。

代理服务技术基于应用层，需要检查数据包的内容，能够对基于高层协议的攻击进行拦截，安全性较包过滤技术要好。缺点是处理速度比较慢，不适用于高速网之间的应用。

另外，代理使用一个客户程序与特定的中间节点连接，然后中间节点与代理服务器进行实际连接。因此，使用这类防火墙时外部网络与内部网络之间不存在直接连接，即使防火墙发生了问题，外部网络也无法与被保护的网络连接。

【答案】B

1.6.5　加密技术和认证技术

1. 加密技术

加密技术是最常用的安全保密手段，数据加密技术的关键在于加密/解密算法和密钥管理。加密技术包括两个元素：算法和密钥。数据加密的基本过程就是对原来为明文的文件或数据按某种加密算法进行处理，使其成为不可读的一段代码，通常称为“密文”。密文只能在输入相应的密钥之后才能显示出原来的内容，通过这样的途径保护数据不被窃取。一般将数据加密技术分成两类：对称加密和非对称加密。

1)　对称加密技术

对称加密的体制模型如图 1-6 所示。其特点是文件加密和解密使用相同的密钥。

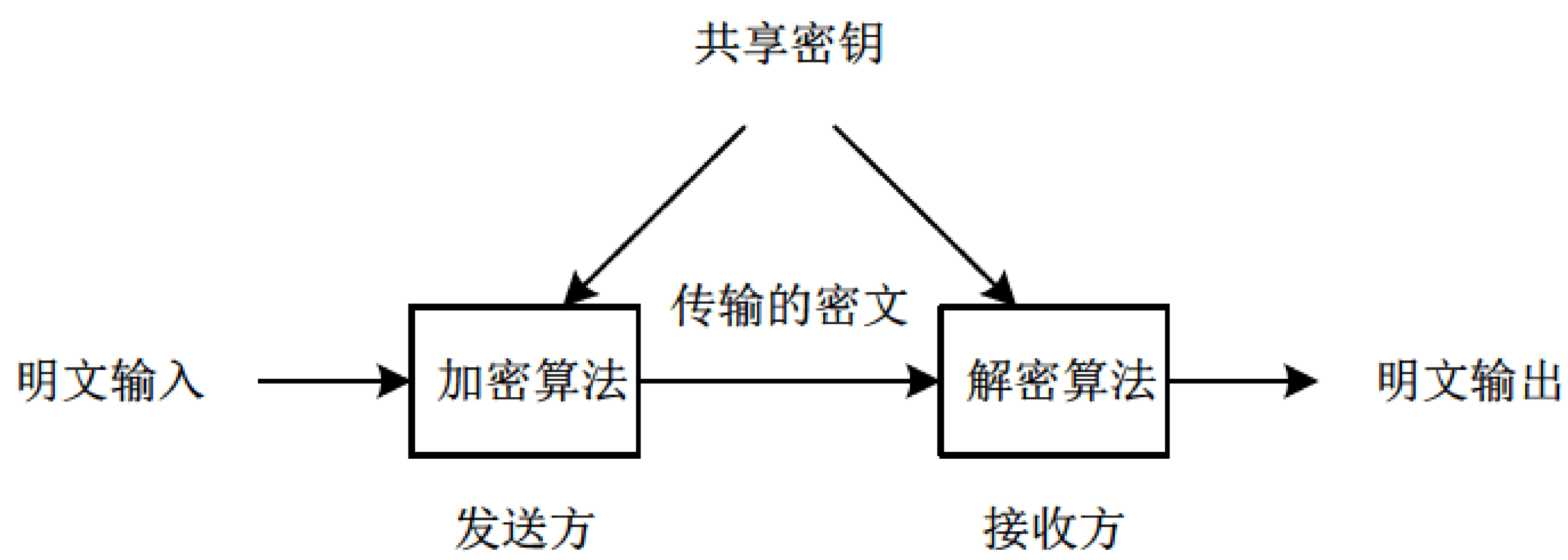


图 1-6　对称加密的体制模型

常用的对称加密算法有如下几个。

- 数据加密标准(Digital Encryption Standard，DES)算法。
- 三重 DES(3DES 或称 TDES)。
- RC-5(Rivest Cipher 5)。
- 国际数据加密算法(International Data Encryption Algorithm，IDEA)。

2)　非对称加密技术

与对称加密算法不同，非对称加密算法需要两个密钥：公开密钥和私有密钥(即“公钥”和“私钥”)。这里的“公钥”是指可以对外公布的，“私钥”只能由持有人知道。因为加密和解密使用的是两个不同的密钥，所以这种算法称为非对称加密算法(或公钥加密算法)。非对称加密有两个不同的体制，如图 1-7 所示。

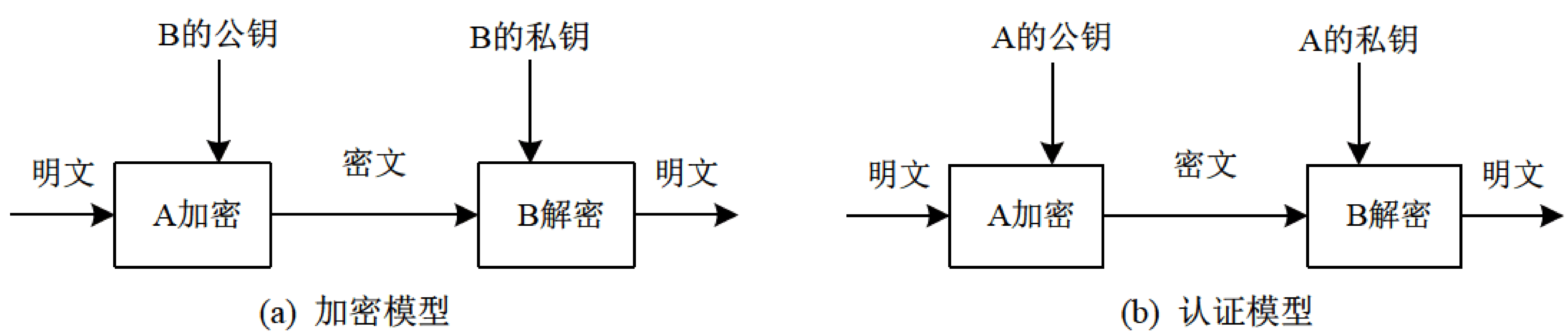


图 1-7　非对称加密的体制模型

非对称加密算法实现机密信息交换的基本过程是：甲方生成一对密钥并将其中的一把作为公用密钥向其他方公开；得到该公用密钥的乙方使用该密钥对机密信息进行加密后再发送给甲方；甲方再用自己保存的另一把专用密钥对加密后的信息进行解密。甲方只能用其专用密钥解密由其公用密钥加密后的信息。RSA(Rivest Shamir and Adleman)是一种典型的非对称加密算法。

真题链接

【例 1-18】下面算法中，不属于公开密钥加密算法的是__(9)__。(2013 年 11 月真题 9)

(9)　A．ECC　　B．DSA　　C．RSA　　D．DES

【解析】ECC(Elliptic Curves Cryptography)：椭圆曲线密码编码学。RSA：由 RSA 公司发明，是一个支持变长密钥的公共密钥算法，需要加密的文件块的长度也是可变的；DSA(Digital Signature Algorithm)：

数字签名算法，是一种标准的 DSS(数字签名标准)。这三种都属于公开密钥加密算法。

DES：数据加密标准，速度较快，适用于加密大量数据的场合，是一种对称加密算法。

【答案】D

2. 认证技术

1) 认证技术概述

认证技术主要解决网络通信过程中通信双方的身份认证问题。认证的过程涉及加密和密钥交换。通常，加密可使用对称加密、不对称加密及两种加密方法的混合方法。认证一般有账户名/口令认证、使用摘要算法认证、基于 PKI(Public Key Infrastructure，公开密钥体系)的认证等几种方法。一个有效的 PKI 系统必须是安全的和透明的，用户在获得加密和数字签名服务时，不需要详细了解 PKI 的内部运行机制。

2) 哈希函数与信息摘要

哈希(Hash，又称杂凑或散列)函数提供了这样一种计算过程：输入一个长度不固定的字符串，返回一串定长度的字符串(又称 Hash 值)。单向 Hash(散列)函数用于产生信息摘要。

信息摘要简要地描述了一份较长的信息或文件，它可以被看作一份长文件的“数字指纹”。信息摘要用于创建数字签名。

3) 数字签名

数字签名是通过一个单向散列函数对要传送的报文进行处理得到的，用以认证报文来源并核实报文是否发生变化的一个字母数字串。数字签名可以解决否认、伪造、篡改及冒充等问题，应用范围十分广泛，比如加密信件、商务信函、订货购买系统、远程金融交易、自动模式处理等。

数字签名和数字加密的过程虽然都使用公开密钥体系，但实现的过程正好相反，使用的密钥对也不同。数字签名使用的是发送方的密钥对，发送方用自己的私有密钥进行加密，接收方用发送方的公开密钥进行解密，这是一对多的关系，任何拥有发送方公开密钥的人都可以验证数字签名的正确性。数字加密使用的是接收方的密钥对，这是多对一的关系，任何知道接收方公开密钥的人都可以向接收方发送加密信息，只有唯一拥有接收方私有密钥的人才能对信息解密。另外，数字签名只采用了非对称密钥加密算法，它能保证发送信息的完整性、身份认证和不可否认性，而数字加密则采用了对称密钥加密算法和非对称密钥加密算法相结合的方法，它能保证发送信息的保密性。

4) SSL 协议

SSL(Secure Sockets Layer)协议是网景(Netscape)公司提出的基于 Web 应用的安全协议。

SSL 协议主要提供三方面的服务：用户和服务器的合法性认证，加密数据以隐藏被传送的数据，保护数据的完整性，目的是在两个通信应用程序之间提供私密性和可靠性。对于电子商务应用来说，使用 SSL 可以保证信息的真实性、完整性和保密性。

5) 数字时间戳技术

数字时间戳技术是数字签名技术的一种变种。数字时间戳服务(Digital Time-stamp Service，DTS)是网上电子商务提供的安全服务项目之一，能提供电子文件的日期和时间信

息的安全保护。

时间戳(Time-stamp)是一个经过加密后形成的凭证文档，它包括以下三个部分。

- 需加时间戳的文件的摘要。
- DTS 收到文件的日期和时间。
- DTS 的数字签名。

一般来说，时间戳产生的过程为：用户首先将需要时间戳的文件用 Hash 编码加密形成摘要，然后将该摘要发送到 DTS，DTS 在加入收到文件摘要的日期和时间信息后再对该文件加密(数字签名)，然后送回用户。

真题链接

【例 1-19】用户 B 收到用户 A 带数字签名的消息 M，为了验证 M 的真实性，首先需要从 CA 获取用户 A 的数字证书，并利用__(8)__验证该证书的真伪，然后利用__(9)__验证 M 的真实性。(2012 年 11 月真题 8～9)

(8) A．CA 的公钥　　B．B 的私钥　　C．A 的公钥　　D．B 的公钥

(9) A．CA 的公钥　　B．B 的私钥　　C．A 的公钥　　D．B 的公钥

【解析】数字证书是一个经证书认证中心(CA)数字签名的包含公开密钥拥有者信息以及公开密钥的文件。要验证证书的真伪，可利用 CA 的公钥验证 CA 的数字签名。数字证书采用公钥体制，即利用一对互相匹配的密钥进行加密、解密。每个用户自己设定一个特定的仅为本人所知的私有密钥(私钥)，用它进行解密和签名；同时设定一个公共密钥(公钥)并由本人公开，为一组用户所共享，用于加密和验证签名。

【答案】(8)A；(9)C

1.7　可靠性与系统性能评测基础知识

1.7.1　计算机可靠性

1. 计算机可靠性概述

计算机系统的可靠性是指从它开始运行(t=0)到某个时刻 t 这段时间内能正常运行的概率，用 $R(t)$表示。

失效率是指单位时间内失效的元件数与元件总数的比例，用λ表示，当λ为常数时，可靠性与失效率的关系为 $R(t)=e^{-\lambda t}$ 。

两次故障之间系统能正常工作的时间的平均值称为平均无故障时间(MTBF)，即 MTBF=1/λ。

通常用平均修复时间(MTRF)来表示计算机的可维修性，即计算机的维修效率，指从故障发生到机器修复平均所需要的时间。计算机的可用性是指计算机的使用效率，它以系统在执行任务的任意时刻能正常工作的概率 A 来表示，即

$$A=\frac{\text{MTBF}}{\text{MTBF}+\text{MTRF}}$$

计算机的 RAS 技术是指用可靠性 R、可用性 A 和可维修性 S 这三个指标衡量一个计算机系统。但在实际应用中，引起计算机故障的原因除了元器件以外，还与组装工艺、逻辑设计等因素有关。

2. 计算机可靠性模型

常见的计算机系统可靠性数学模型如下。

- 串联系统。可靠性 $R = R_1 R_2 \cdots R_N$，失效率 $\lambda = \lambda_1 + \lambda_2 + \cdots + \lambda_N$。
- 并联系统。可靠性 $R = 1 - (1 - R_1) \times (1 - R_2) \times \cdots \times (1 - R_N)$，失效率为

$$\mu = \frac{1}{\frac{1}{\lambda}\sum_{j=1}^{N}\frac{1}{j}}$$

- N 模冗余系统。可靠性为

$$R = \sum_{i=n+1}^{N}\binom{i}{N} \times R_0^i (1 - R_0)^{N-i}$$，其 $\binom{i}{N}$ 表示从 N 个元素中取 i 个元素的组合数。

提高计算机的可靠性一般采取两项措施：提高元器件质量，改进加工工艺与工艺结构，完善电路设计；发展容错技术。

1.7.2 计算机系统的性能评价

1. 性能评测常用方法

性能评测常用方法如下。

- 时钟频率。
- 指令执行速度。
- 等效指令速度法。
- 数据处理速率(PDR)法。
- 核心程序法。

2. 基准测试程序

基准测试程序法是目前一致承认的测试性能较好的方法，有多种多样的基准程序，如主要测试整数性能的基准程序、测试浮点性能的基准程序等。常见的基准测试程序有如下四种。

- 整数测试程序。
- 浮点测试程序。
- SPEC(System Performance Evaluation Cooperation，系统性能评估机构)基准测试程序。
- TPC(Transaction Processing Performance Council，事务处理委员会)基准程序。

1.8　应试加油站

1.8.1　考频统计

通过对近几年考试真题的分析，“计算机系统知识”模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 1-2 所示。

表 1-2　历年考题知识点分布统计表

年　份	题　号	知 识 点	分值/分
2014 年 11 月	上午：1～9	总线系统、CPU、存储器、精简指令集计算机、病毒、防火墙	9
	下午：无		0
2014 年 5 月	上午：1～9	CPU、机器字长符号位、海明码、内存、寻址方式、病毒、防火墙、容错系统	9
	下午：无		0
2013 年 11 月	上午：1～9	Cache 地址映像、指令寄存器、逻辑运算、指令流水线、RISC 体系、寻址范围、DoS 攻击、PKI 体制、加密算法	9
	下午：无		0
2013 年 5 月	上午：1～9	存储器、中断、DMA、内存容量计算、认证技术、防火墙、病毒	9
	下午：无		0
2012 年 11 月	上午：1～9	CPU、循环冗余校验码、数的表示方法、相联存储器、输入输出技术、系统总线、安全协议、数字证书	9
	下午：无		0
2012 年 5 月	上午：1～9	Cache、内存地址、相联存储器、寻址方式、流水线、数字签名、Windows 用户、身份验证	9
	下午：无		0

纵观历年试卷，本章知识点是以选择题的形式出现在试卷中的。本章知识点在历次考试的上午试卷中，所考查的题量大约为 9 道选择题，所占分值为 9 分（约占试卷总分值 75 分中的 12%）；在下午试卷中没有相关考题。本章考题主要检验考生是否理解相关的理论知识点，考试难度较低。寄存器、数据表示、指令系统、数字证书与签名、网络攻击、计算机病毒等内容是考核重点。

1.8.2　解题技巧

【典型题 1-1】 指令寄存器的位数取决于__(2)__。(2013 年 11 月真题 2)

(2) A．存储器的容量　　B．指令字长

C．数据总线的宽度　　D．地址总线的宽度

【解析】 当 CPU 执行一条指令时，要先把它从内存储器读取到缓冲寄存器中，再送入指令寄存器 IR

暂存，IR 是用来存放要执行的指令的，所以很明显跟指令的长度有关。

【答案】B

【典型题 1-2】若计算机存储数据采用的是双符号位(00 表示正号、11 表示负号)，两个符号相同的数相加时，如果运算结果的两个符号位经 (3) 运算得 1，则可断定这两个数相加的结果产生了溢出。(2013 年 11 月真题 3)

(3) A．逻辑与　　B．逻辑或　　C．逻辑同或　　D．逻辑异或

【解析】本题考查逻辑运算，两个相同符号位经过异或可以得到 1，说明符号位不同，发生了变化，因此可以判断这两个数相加的结果产生了溢出。

【答案】D

【典型题 1-3】在 CPU 中， (1) 不仅要保证指令的正确执行，还要能够处理异常事件。(2012 年 11 月真题 1)

(1) A．运算器　　B．控制器　　C．寄存器组　　D．内部总线

【解析】控制器负责完成协调和指挥整个计算机系统的操作，是发布命令的决策机构。运算器是数据加工部件，负责执行算术运算和逻辑运算。寄存器一般用来保存程序的中间结果，为随后的指令快速提供操作数，从而避免把中间结果存入内存，再读取内存的操作。内部总线将处理器内部的所有结构单元相连。

【答案】B

【典型题 1-4】循环冗余校验码(CRC)利用生成多项式进行编码。设数据位为 k 位，校验位为 r 位，则 CRC 码的格式为 (2) 。(2012 年 11 月真题 2)

(2) A．k 个数据位之后跟 r 个校验位　　B．r 个校验位之后跟 k 个数据位

C．r 个校验位随机加入 k 个数据位中　　D．r 个校验位等间隔地加入 k 个数据位中

【解析】循环冗余校验码利用生成多项式为 k 个数据位产生 r 个校验位来进行编码，其编码长度为 $k+r$。编码的格式为：

k	$k-1$	…	2	1	r	$r-1$	…	2	1
←数据位→					←校验位→				

【答案】A

【典型题 1-5】以下关于数的定点表示和浮点表示的叙述中，不正确的是 (3) 。(2012 年 11 月真题 3)

(3) A．定点表示法表示的数(称为定点数)常分为定点整数和定点小数两种

B．定点表示法中，小数点需要占用一个存储位

C．浮点表示法用阶码和尾数来表示数，称为浮点数

D．在总位数相同的情况下，浮点表示法可以表示更大的数

【解析】定点数是小数点位置固定不变的数。定点数分定点整数和定点小数，前者约定小数点在最低有效值位之后，后者约定小数点在最高有效值位之前。可见，在定点表示法中，小数点不需要占用存储位。

【答案】B

【典型题 1-6】若用 256K × 8bit 的存储器芯片，构成地址 40000000H 到 400FFFFFH 且按字节编址的内存区域，则需 (6) 片芯片。(2014 年 5 月真题 6)

(6) A. 4　　B. 8　　C. 16　　D. 32

【解析】内存区域从 40000000H 到 400FFFFFH，占用的字节数为

$$400FFFFFH-40000000H+1=100000H=1\ 0000\ 0000\ 0000\ 0000\ 0000\ B=2^{20}=1024K$$

一片 256K×8bit 的存储器芯片的存储容量为 256KB，需要的芯片数为 1024÷256=4。

【答案】A

【典型题 1-7】常用的虚拟存储器由__(1)__两级存储器组成。(2013 年 5 月真题 1)

(1)　A. 主存—辅存　　B. 主存—网盘　　C. Cache—主存　　D. Cache—硬盘

解　析：虚拟存储器是为了给用户提供更大的随机存取空间而采用的一种存储技术。它将内存与外存结合使用，好像有一个容量极大的内存储器，工作速度接近于主存，每位成本又与辅存相近，在整机形成多层次存储系统。所以虚拟存储器由主存和辅存两级存储器组成。

【答案】A

【典型题 1-8】地址编号从 80000H 到 BFFFFH 且按字节编址的内存容量为__(5)__KB，若用 16K×4bit 的存储器芯片构成该内存，共需__(6)__片。(2013 年 5 月真题 5～6)

(5)　A. 128　　B. 256　　C.512　　D. 1024

(6)　A. 8　　B. 16　　C.32　　D.64

解　析：从80000H到BFFFFH有BFFFFH−80000H+1个地址单元，即40000H个地址单元。若按字节编址，有2^{18}个字节，即256KB。若用16K×4bit的存储芯片，需要(256K×2×4bit)/(16K×4bit)=32片芯片。

【答案】(5)B、(6)C

【典型题 1-9】__(4)__不属于按寻址方式划分的一类存储器。(2012 年 11 月真题 4)

(4)　A. 随机存储器　　B. 顺序存储器　　C. 相联存储器　　D. 直接存储器

【解析】相联存储器是按内容访问的一种存储器，是按访问方式划分的。存储器按访问方式可分为按地址访问的存储器和按内容访问的存储器。

【答案】C

【典型题 1-10】位于 CPU 与主存之间的高速缓冲存储器 Cache 用于存放部分主存数据的拷贝，主存地址与 Cache 地址之间的转换工作由__(1)__完成。(2012 年 5 月真题 1)

(1)　A. 硬件　　B. 软件　　C. 用户　　D. 程序员

【解析】主存地址与 Cache 地址之间的转换是与主存块、Cache 块之间的映射关系紧密联系的。而这个转换工作是由硬件完成的。该过程对程序员是透明的。

【答案】A

【典型题 1-11】内存单元按字节编址，地址 0000A000H～0000BFFFH 共有__(2)__个存储单元。(2012 年 5 月真题 2)

(2)　A. 8192K　　B. 1024K　　C. 13K　　D. 8K

【解析】存储单元的个数0000BFFFH−0000A000H+1=00001FFFH+1=00002000H=2^{13}=8K。

【答案】D

【典型题 1-12】相联存储器按__(3)__访问。(2012 年 5 月真题 3)

(3)　A. 地址　　B. 先入后出的方式　　C. 内容　　D. 先入先出的方式

【解析】相联存储器是一种按内容访问的存储器。其工作原理是把数据或数据的某一部分作为关键字，将该关键字与存储器中的每一个单元进行比较，找出存储器中所有与关键字相同的数据。

【答案】C

【典型题 1-13】为了便于实现多级中断嵌套，使用__(3)__来保护断点和现场最有效。(2013 年 5 月真题 3)

(3) A. ROM　　B. 中断向量表　　C. 通用寄存器　　D. 堆栈

【解析】多重中断时，每次中断出现的断点都必须保存起来。中断系统对断点的保存都是在中断周期内由中断隐指令实现的，对用户是透明的。断点可以保存在堆栈中，由于堆栈具有先进后出的特点，所以依次将程序的断点压入堆栈中。出栈时，按相反顺序便可准确返回到程序间断处。

【答案】D

【典型题 1-14】DMA 工作方式下，在__(4)__之间建立了直接的数据通路。(2013 年 5 月真题 4)

(4) A. CPU 与外设　　B. CPU 与主存　　C. 主存与外设　　D. 外设与外设

【解析】在 DMA 模式下，CPU 只需向 DMA 控制器下达指令，让 DMA 控制器来处理数据的传送，数据传送完毕再把信息反馈给 CPU，这样就在很大程度上减少了 CPU 资源的占有率。DMA 请求只是使 CPU 暂停一下，不需要对断点和现场的处理，并且是由 DMA 控制外设与主存之间的数据传送，无需 CPU 的干预，因此 DMA 工作方式下，在主存与外设之间建立了直接的数据通路。

【答案】C

【典型题 1-15】在 I/O 设备与主机间进行数据传输时，CPU 只需在开始和结束时作少量处理，而无须干预数据传送过程的是__(5)__方式。(2012 年 11 月真题 5)

(5) A. 中断　　B. 程序查询　　C. 无条件传送　　D. 直接存储器存取

【解析】CPU 与外设之间交换数据的方式有直接程序控制方式、中断方式、直接存储器存取方式和通道控制方式。

直接程序控制方式分为：无条件传送和程序查询。在无条件传送情况下，外设总是准备好的，外设即可以无条件地随时接收 CPU 输出的数据，也可无条件地随时向 CPU 输入数据。在程序查询方式下，则需要 CPU 执行程序查询外设的状态，判断外设是否准备好了。

中断方式下，数据传送的基本过程是：当 I/O 系统与外设交换数据时，CPU 无须等待，也不必查询 I/O 的状态，可以去处理其他任务。当 I/O 系统完成了数据传输后则向 CPU 发送中断信号，然后 CPU 保存正在执行的现场，转入 I/O 中断服务程序，完成与 I/O 系统的数据交换，再返回原程序继续执行。

可见，不论是无条件传送、利用查询方式传送还是利用中断方式传送，都需要由 CPU 通过执行程序来实现，限制了数据的传输速度。

直接存储器存取方式则是通过硬件实现主存与 I/O 设备间的直接数据传送，在数据传送的过程中不需要 CPU 的任何干涉，只需要 CPU 在过程开始时和在过程结束时的处理。

【答案】D

【典型题 1-16】__(6)__不属于系统总线。(2012 年 11 月真题 6)

(6) A. ISA　　B. EISA　　C. SCSI　　D. PCI

【解析】系统总线又称内总线，其速度对系统性能有极大的影响。常见的系统总线有 ISA 总线、EISA 总线、PCI 总线。SCSI 总线是并行外总线，广泛用于连接软硬磁盘、光盘和扫描仪等。

【答案】C

【典型题 1-17】某指令流水线由 4 段组成，各段所需要的时间如图 1-8 所示。连续输入 8 条指令时的吞吐率(单位时间内流水线所完成的任务数或输出的结果数)为__(4)__。(2013 年 11 月真题 4)

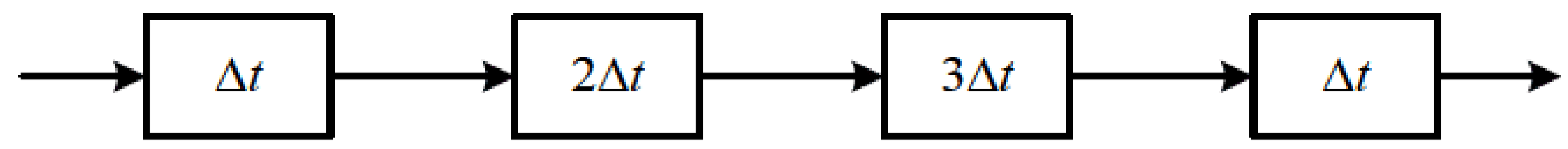


图 1-8　某指令流水线时序

(4)A．8/56Δ*t*　　B．8/32Δ*t*　　C．8/28Δ*t*　　D．8/24Δ*t*

【解析】吞吐率是指单位时间里流水线处理机流出的结果数。对指令而言，就是单位时间里执行的指令数。如果流水线的子过程所用时间不一样，则一条指令的吞吐率 p 应为最长子过程的倒数：

$$p = 1/\max\{\Delta t_1, \Delta t_2, \cdots, \Delta t_m\}$$

所以 4 段流水线中最长的时间段为 3Δ*t*，8 条指令占 24Δ*t*，到达下一个流水线中经过 4 个Δ*t*，执行 8 条指令的时间为 28Δ*t* 所以总的吞吐率为 8/28Δ*t*。

【答案】C

【典型题 1-18】__(5)__不是 RISC 的特点。(2013 年 11 月真题 5)

(5) A．指令种类丰富　　B．高效的流水线操作

C．寻址方式较少　　D．硬布线控制

【解析】RISC 的关键技术如下。

① 重叠寄存器窗口技术。在伯克利的 RISC 项目中，首先采用了重叠寄存器窗口(Overlapping Register Windows)技术。

② 优化编译技术。RISC 使用了大量的寄存器，如何合理分配寄存器、提高寄存器的使用效率及减少访存次数等，都应通过编译技术的优化来实现。

③ 超流水及超标量技术。这是 RISC 为了进一步提高流水线速度而采用的技术。

④ 硬布线逻辑与微程序在微程序技术中相结合。

【答案】A

【典型题 1-19】若某计算机字长为 32 位，内存容量为 2GB，按字编址，则可寻址范围为__(6)__。(2013 年 11 月真题 6)

(6) A．1024M　　B．1GB　　C．512MB　　D．2GB

【解析】计算机字长为 32 位，一个字长为 4 个字节，即每个存储字包含 4 个可寻址的字节。故可寻址范围为 2GB / 4=512M。

【答案】C

【典型题 1-20】若 CPU 要执行的指令为 MOV R1,#45(即将数值 45 传送到寄存器 R1 中)，则该指令中采用的寻址方式为__(4)__。(2012 年 5 月真题 4)

(4) A. 直接寻址和立即寻址　　B. 寄存器寻址和立即寻址

C. 相对寻址和直接寻址　　D. 寄存器间接寻址和直接寻址

【解析】操作数作为指令的一部分而直接写在指令中为立即寻址。本题中数值 45 直接包含在指令中，为立即寻址。把目标操作数存入寄存器则为寄存器寻址，本题中，需要将数值 45 传送到寄存器 R1 中，为寄存器寻址。

【答案】B

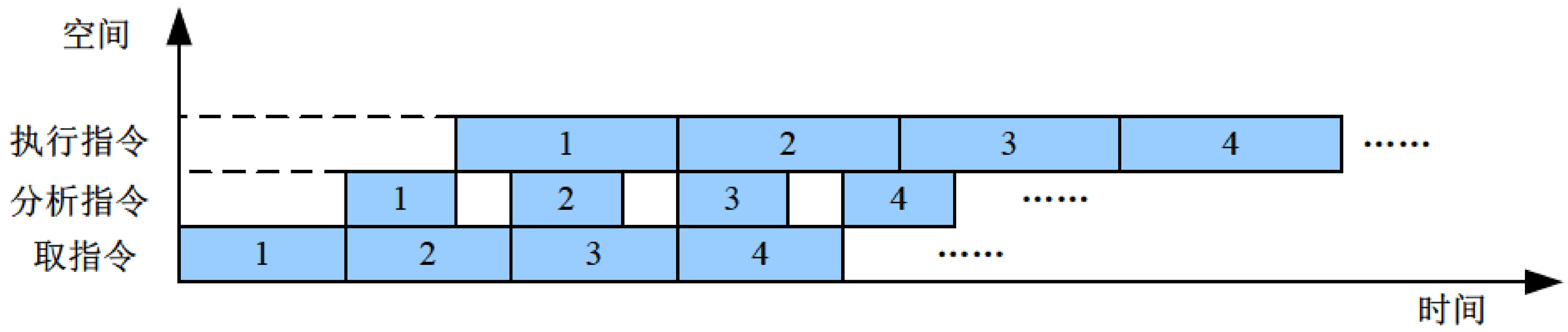
【典型题 1-21】一条指令的执行过程可以分解为取指、分析和执行三步，在取指时间 $t_{取指}$=3Δ*t*、分析时间 $t_{分析}$=2Δ*t*、执行时间 $t_{执行}$=4Δ*t* 的情况下，若按串行方式执行，则 10 条指令全部执行完需要__(5)__Δ*t*；若按照流水方式执行，则执行完 10 条指令需要__(6)__Δ*t*。(2012 年 5 月真题 5～6)

(5) A. 40　　B. 70　　C. 90　　D. 100

(6) A. 20　　B. 30　　C. 40　　D. 45

【解析】执行一条指令的时间为取指时间、分析时间、执行时间之和，为 9Δ*t*。按串行方式执行，执

行完一条指令之后才能执行下一条指令，因此执行 10 条指令的时间为 $9\Delta t\times10=90\Delta t$。按流水方式执行的过程如下，由于执行指令的时间最长，执行 10 条指令的时间为 $10t_{执行}+t_{取指}+t_{分析}=4\Delta t\times10+3\Delta t+2\Delta t=45\Delta t$。



【答案】(5)C；(6)D

【典型题 1-22】防火墙的工作层次是决定防火墙效率及安全的主要因素，下面的叙述中正确的是__(8)__。(2014 年 5 月真题 8)

(8) A. 防火墙工作层次越低，则工作效率越高，同时安全性越高
B. 防火墙工作层次越低，则工作效率越低，同时安全性越低
C. 防火墙工作层次越高，则工作效率越高，同时安全性越低
D. 防火墙工作层次越高，则工作效率越低，同时安全性越高

【解析】防火墙工作层次越高，实现过程复杂，对数据包的理解力越好，对非法包的判断能力越高，但工作效率越低；防火墙工作层次越低，实现过程越简单，其工作效率越高，同时安全性越差。

【答案】D

【典型题 1-23】利用报文摘要算法生成报文摘要的目的是__(7)__。(2013 年 5 月真题 7)

(7) A. 验证通信对方的身份，防止假冒　　B. 对传输数据进行加密，防止数据被窃听
C. 防止发送方否认发送过数据　　D. 防止发送的报文被篡改

【解析】报文摘要是用来保证数据完整性的。传输的数据一旦被修改那么计算出的摘要就不同，只要对比两次摘要就可确定数据是否被修改过。因此其目的是为了防止发送的报文被篡改。

【答案】D

【典型题 1-24】防火墙通常分为内、外网和 DMZ 三个区域，按照受保护程度，从高到低正确的排列次序为__(8)__。(2013 年 5 月真题 8)

(8) A. 内网、外网和 DMZ　　B. 外网、内网和 DMZ
C. DMZ、内网和外网　　D. 内网、DMZ 和外网

【解析】DMZ 是为了解决安装防火墙后外部网络不能访问内部网络服务器的问题，而设立的一个非安全系统与安全系统之间的缓冲区，这个缓冲区位于企业内部网络和外部网络之间的小网络区域内。所以按照受保护程度从高到低就显而易见了。

【答案】D

【典型题 1-25】今年来，在我国出现的各种病毒中，__(9)__病毒通过木马形式感染智能手机。(2013 年 5 月真题 9)

(9) A. 欢乐时光　　B. 熊猫烧香　　C. X 卧底　　D. CIH

【解析】“欢乐时光”是一个 VB 源程序病毒，专门感染.htm、.html、.vbs、.asp 和.htt 文件。它作为电子邮件的附件，并利用 Outlook Express 的性能缺陷把自己传播出去，即利用 Microsoft Outlook Express 的一个安全漏洞，可以在用户没有运行任何附件时就自动运行。“欢乐时光”病毒还可以利用 Outlook Express 的信纸功能，使自己复制在信纸的 Html 模板上，以便传播。

“熊猫烧香”其实是一种蠕虫病毒的变种，而且是经过多次变种而来的，由于中毒电脑的可执行文件会出现“熊猫烧香”图案，所以也被称为“熊猫烧香”病毒。但原病毒只会对 EXE 图标进行替换，并不会对系统本身进行破坏。

“X 卧底”软件是一种安装在手机里的监控软件，若在手机里安装了这种软件，则该手机的所有短信、通话记录都将自动上传到后台服务器，安装者登录后台服务器便可看见目标手机所收发的信息及通话内容，因此“X 卧底”病毒通过木马形式感染智能手机。

CIH 病毒是一种能够破坏计算机系统硬件的恶性病毒。

【答案】C

【典型题 1-26】下列网络攻击行为中，属于 DoS 攻击的是__(7)__。(2013 年 11 月真题 7)

(7) A．特洛伊木马攻击　　B．SYN Flooding 攻击
C．端口欺骗攻击　　D．IP 欺骗攻击

【解析】DoS 即拒绝服务，其攻击目的是使计算机或网络无法提供正常的服务。最常见的 DoS 攻击有计算机网络带宽攻击和连通性攻击。

特洛伊木马是一种秘密潜伏的能够通过远程网络进行控制的恶意程序。控制者可以控制被秘密植入木马的计算机的一切动作和资源，是恶意攻击者窃取信息的工具。特洛伊木马没有复制能力，它的特点是伪装成一个实用工具或者一个可爱的游戏，这会诱使用户将其安装在 PC 或者服务器上。

SYNFlooding 攻击：我们知道当用户进行一次标准的 TCP(Transmission Control Protocol，传输控制协议)连接时，会有一个 3 次握手过程。它是一种 DoS 攻击。

IP 欺骗攻击是伪造数据包源 IP 地址的攻击。

【答案】B

【典型题 1-27】PKI 体制中，保证数字证书不被篡改的方法是__(8)__。(2013 年 11 月真题 8)

(8) A．用 CA 的私钥对数字证书签名　　B．用 CA 的公钥对数字证书签名
C．用证书主人的私钥对数字证书签名　　D．用证书主人的公钥对数字证书签名

【解析】PKI(Public Key Infrastructure)是基于公开密钥理论和技术建立起来的安全体系，如果数据在传输和处理过程中被篡改，接收方就不会收到正确的数字签名。为了保证数字证书不被篡改，采用 CA 的私钥对数字证书签名。

【答案】A

【典型题 1-28】下列安全协议中，与 TLS 最接近的协议是__(7)__。(2012 年 11 月真题 7)

(7) A．PGP　　B．SSL　　C．HTTPS　　D．IPSec

【解析】安全传输层协议(TLS)用于在两个通信应用程序之间提供保密性和数据完整性。该协议由两层组成：TLS 记录协议(TLS Record)和 TLS 握手协议(TLS Handshake)。较低的层为 TLS 记录协议，位于某个可靠的传输协议(例如 TCP)上面。

SSL(Secure Sockets Layer，安全套接层)及其继任者 TLS(Transport Layer Security)是为网络通信提供安全及数据完整性的一种安全协议。TLS 与 SSL 在传输层对网络连接进行加密。SSL 协议位于 TCP/IP 协议与各种应用层协议之间，为数据通信提供安全支持。SSL 协议可分为两层：①SSL 记录协议，它建立在可靠的传输协议(如 TCP)之上，为高层协议提供数据封装、压缩、加密等基本功能的支持；②SSL 握手协议，它建立在 SSL 记录协议之上，用于在实际的数据传输开始前，通信双方进行身份认证、协商加密算法、交换加密密钥等。

PGP(Pretty Good Privacy)是一个基于 RSA 公钥加密体系的邮件加密软件。可以用它对邮件加密以防止非授权者阅读，它还能为邮件加上数字签名从而使收信人可以确认邮件的发送者，并能确信邮件没有被篡改。

HTTPS 是以安全为目标的 HTTP 通道。HTTPS 的安全基础是 SSL。

“Internet 协议安全性(IPSec)”是一种开放标准的框架结构，通过使用加密的安全服务以确保在 Internet 协议(IP)网络上进行保密而安全的通信。

【答案】B

【典型题 1-29】甲和乙要进行通信，甲对发送的消息附加了数字签名，乙收到该消息后利用__(7)__验证该消息的真实性。(2012 年 5 月真题 7)

(7) A. 甲的公钥　　B. 甲的私钥　　C. 乙的公钥　　D. 乙的私钥

【解析】数字签名使用公开密钥体系，签名时使用发送方的密钥对，发送方用自己的私有密钥进行加密，接收方用发送方的公开密钥进行解密。任何拥有发送方公开密钥的人都可以验证数字签名的真实性。

【答案】A

【典型题 1-30】在 Windows 系统中，默认权限最低的用户组是__(8)__。(2012 年 5 月真题 8)

(8) A. everyone　　B. administrators　　C. power users　　D. users

【解析】administrators 组内的用户，都具备系统管理员的权限，它们拥有对这台计算机最大的控制权限，可以执行整台计算机的管理任务。users 组员只拥有一些基本的权利，例如运行应用程序，但是他们不能修改操作系统的设置，不能更改其他用户的数据，不能关闭服务器级的计算机。所有添加的本地用户账户自动属于该组。power users 组内的用户具备比 users 组更多的权利，但是比 administrators 组拥有的权利要少一些。everyone 是个抽象组，任何一个用户都属于这个组。

【答案】A

【典型题 1-31】IIS 6.0 支持的身份验证安全机制有 4 种验证方法，其中安全级别最高的验证方法是__(9)__。(2012 年 5 月真题 9)

(9) A. 匿名身份验证　　B. 集成 Windows 身份验证
C. 基本身份验证　　D. 摘要式身份验证

【解析】①匿名身份验证，不验证访问用户的身份，客户端不需要提供任何身份验证的凭据，服务端把这样的访问作为匿名的访问，并把这样的访问用户都映射到一个服务端的账户。②基本身份验证，完全是把用户名和密码用明文传送到服务端验证，服务器直接验证服务器本地是否有用户跟客户端提供的用户名和密码相匹配，如果有则通过验证。③摘要式身份验证的设计目的是提高基本身份验证系统所提供的安全性，它允许服务器指定客户端应该使用的加密或散列机制，并提供了在算法中注入随机明文以协助防止重播攻击的选项。④集成 Windows 身份验证可以使用 NTLM 或 Kerberos v5 身份验证，当 Internet Explorer 试图设为集成验证的 IIS 的资源时，IIS 发送两个 WWW 身份验证头：Negotiate 和 NTLM。⑤基本身份验证的安全级别较低，摘要式身份验证的安全级别中等，集成 Windows 身份验证的安全级别较高。

【答案】B

1.9 过关习题

1. 在 CPU 的寄存器中，______对用户是完全透明的。

A. 程序计数器　　B. 指令寄存器

C. 状态寄存器　　D. 通用寄存器

2. CPU 中译码器的主要作用是进行______。

A. 地址译码　　B. 指令译码

C. 数据译码　　D. 选择多路数据至 ALU

3. 在 CPU 中用于跟踪指令地址的寄存器是______。

A. 地址寄存器(MAR)　　B. 数据寄存器(MDR)

C. 程序计数器(PC)　　D. 指令寄存器(IR)

4. 原码表示法和补码表示法是计算机中用于表示数据的两种编码方法，在计算机系统中常采用补码来表示和运算数据，原因是采用补码可以______。

A. 保证运算过程与手工运算方法保持一致

B. 简化计算机运算部件的设计

C. 提高数据的运算速度

D. 提高数据的运算精度

5. 若某计算机采用 8 位整数补码表示数据，则运算______将产生溢出。

A. −127+1　　B. −127−1　　C. 127+1　　D. 127−1

6. 计算机中的浮点数由三部分组成：符号位 S、指数部分 E(称为阶码)和尾数部分 M。在总长度固定的情况下，增加 E 的位数、减少 M 的位数可以______。

A. 扩大可表示的数的范围同时降低精度

B. 扩大可表示的数的范围同时提高精度

C. 减小可表示的数的范围同时降低精度

D. 减小可表示的数的范围同时提高精度

7. 以下关于校验码的叙述中，正确的是______。

A. 海明码利用多组数位的奇偶性来检错和纠错　　B. 海明码的码距必须大于等于 1

C. 循环冗余校验码具有很强的检错和纠错能力　　D. 循环冗余校验码的码距必定为 1

8. 若某条无条件转移汇编指令采用直接寻址，则该指令的功能是将指令中的地址码送入______。

A. PC(程序计数器)　　B. AR(地址寄存器)

C. AC(累加器)　　D. ALU(算术逻辑单元)

9. 指令系统中采用不同寻址方式的目的是______。

A. 提高从内存获取数据的速度

B. 提高从外存获取数据的速度

C. 降低操作码的译码难度

D. 扩大寻址空间并提高编程灵活性

10. 某指令的流水线由 5 段组成，第 1、3、5 段所需时间为Δt，第 2、4 段所需时间分别为 $3\Delta t$、$2\Delta t$，如图 1-9 所示，那么连续输入 n 条指令时的吞吐率(单位时间内执行的指令个数)TP 为______。

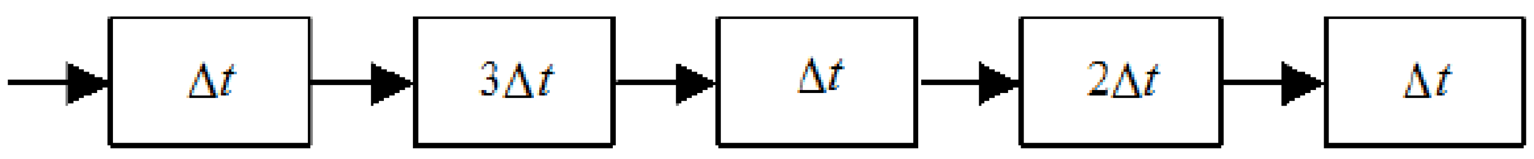


图 1-9　指令流水线

A. $\frac{n}{5\times(3+2)\Delta t}$　　B. $\frac{n}{(3+3+2)\Delta t+3(n-1)\Delta t}$

C. $\frac{n}{(3+2)\Delta t+(n-3)\Delta t}$　　D. $\frac{n}{(3+2)\Delta t+5\times 3\Delta t}$

11. 设用 2K×4 位的存储器芯片组成 16K×8 位的存储器(地址单元为 0000H～3FFFH，每个芯片的地址空间连续)，则地址单元 0BIFH 所在芯片的最小地址编号为______。

A. 0000H　B. 0800H　C. 2000H　D. 2800H

12. 在程序的执行过程中，Cache 与主存的地址映像由______。

A. 专门的硬件自动完成　B. 程序员进行调度

C. 操作系统进行管理　D. 程序员和操作系统共同协调完成

13. 假设某硬盘由 5 个盘片构成(共有 8 个记录面)，盘面有效记录区域的外直径为 30cm，内直径为 10cm，记录位密度为 250 位/mm，磁道密度为 16 道/mm，每磁道分 16 个扇区，每扇区为 512 个字节，则该硬盘的格式化容量约为______MB。

A. $\frac{8\times(30-10)\times10\times250\times16}{8\times1024\times1024}$　　B. $\frac{8\times(30-10)\times10\times16\times16\times512}{2\times1024\times1024}$

C. $\frac{8\times(30-10)\times10\times250\times16\times16}{8\times1024\times1024}$　　D. $\frac{8\times(30-10)\times16\times16\times512}{2\times1024\times1024}$

14. 若某计算机系统的 I/O 接口与主存采用统一编址，则输入输出操作是通过______指令来完成的。

A. 控制　B. 中断　C. 输入输出　D. 访存

15. 在输入输出控制方法中，采用______可以使得设备与主存间的数据块传送无须 CPU 干预。

A. 程序控制输入输出　B. 中断

C. DMA　D. 总线控制

16. 总线复用方式可以______。

A. 提高总线的传输宽度　B. 增加总线的功能

C. 减少总线中信号线的数量　D. 提高 CPU 利用率

17. 在计算机系统中采用总线结构，便于实现系统的积木化构造，同时可以______。

A. 提高数据传输速度　B. 提高数据传输量

C. 减少信息传输线的数量　D. 减少指令系统的复杂性

18. 如果使用大量的连接请求攻击计算机，使得所有可用的系统资源都被消耗殆尽，最终计算机无法再处理合法用户的请求，这种手段属于______攻击。

A. 拒绝服务　B. 口令入侵　C. 网络监听　D. IP 欺骗

19. ARP 攻击造成网络无法跨网段通信的原因是______。

A. 发送大量 ARP 报文造成网络拥塞　B. 伪造网关 ARP 报文使得数据包无法发送到网关

C. ARP 攻击破坏了网络的物理连通性　D. ARP 攻击破坏了网关设备

20. 宏病毒一般感染以______为扩展名的文件。

A. EXE　B. COM　C. DOC　D. DLL

21. 在 IE 浏览器中，安全级别最高的区域设置是______。

A. Internet　B. 本地 Intranet　C. 可信站点　D. 受限站点

22. 通过内部发起连接与外部主机建立联系，由外部主机控制并盗取用户信息的恶意代码为______。

A. 特洛伊木马　B. 蠕虫病毒　C. 宏病毒　D. CIH 病毒

23. 利用______可以获取某 FTP 服务器中是否存在可写目录的信息。

A. 防火墙系统　　B. 漏洞扫描系统　　C. 入侵检测系统　　D. 病毒防御系统

24. 下列选项中，防范网络监听最有效的方法是______。

A. 安装防火墙　　B. 采用无线网络传输　　C. 数据加密　　D. 漏洞扫描

25. 从认证中心(CA)获取用户 B 的数字证书，该证书用______做数字签名；从用户 B 的数字证书中可获得 B 的公钥。

A. CA 的公钥　　B. CA 的私钥　　C. B 的公钥　　D. B 的私钥

26. 用户 A 从 CA 获得用户 B 的数字证书，并利用______验证数字证书的真实性。

A. B 的公钥　　B. B 的私钥　　C. CA 的公钥　　D. CA 的私钥

27. 公钥体系中，私钥用于 (1) ，公钥用于 (2) 。(2010 年 11 月上午试题 66～67)

(1) A. 解密和签名　　B. 加密和签名　　C. 解密和认证　　D. 加密和认证

(2) A. 解密和签名　　B. 加密和签名　　C. 解密和认证　　D. 加密和认证

28. 某计算机系统由图 1-10 所示部件构成，假定每个部件的千小时可靠度为 R，则该系统的千小时可靠度为______。

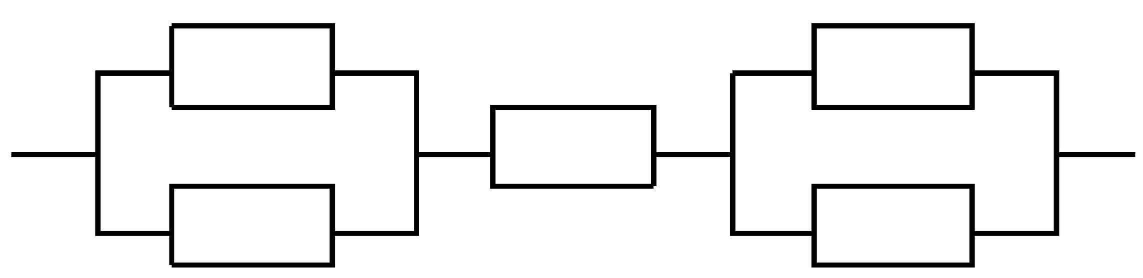


图 1-10　系统模型

A. $R+2R/4$　　B. $R+R^2/4$　　C. $R(1-(1-R)^2)$　　D. $R(1-(1-R)^2)^2$

第 2 章

程序语言基础

本章要点

- 程序设计语言的基本成分
- 汇编语言的基本原理
- 编译程序的基本原理
- 解释程序的基本原理
- 文法和语言的形式描述
- 正规式
- 有限状态机

2.1 程序语言概述

考核说明：本节主要介绍程序设计语言的定义、发展和分类，程序设计语言的基本成分，后者是重点，经常会考到，其中数据成分和函数是考核要点，要理解掌握。

2.1.1 程序设计语言的基本概念

1. 低级语言和高级语言

1) 低级语言

通常称机器语言和汇编语言为低级语言。机器语言是指用 0、1 字符串组成的机器指令序列，是最基本的计算机语言；汇编语言是指用符号表示指令的语言。汇编语言仍然是一种面向机器的语言。

2) 高级语言

高级语言是从人类的逻辑思维角度出发、面向各类应用的程序语言，抽象程度大大提高。这类语言与人们使用的自然语言比较接近，大大提高了程序设计的效率。常见的 C、C++、Java、PHP 等都是高级语言。

2. 编译程序和解释程序

用某种高级语言或汇编语言编写的程序称为源程序，源程序不能直接在计算机上执行。如果源程序是使用汇编语言编写的，则需要一个称为汇编程序的翻译程序将其翻译成目标程序后才能执行。如果源程序是使用某种高级语言编写的，则需要相应的解释程序或编译程序对其进行翻译，然后才能在机器上运行。

注意：在解释方式下，翻译源程序时不生成独立的目标程序，而编译器则会将源程序翻译成独立保存的目标程序。

3. 程序设计语言的定义

(1) 语法：由程序设计语言的基本符号组成程序中的各个语法成分(包括程序)的一组规则，其中由基本符号构成符号(单词)的书写规则称为词法规则，由符号(单词)构成语法成分的规则称为语法规则。程序语言的语法可通过形式语言进行描述。

(2) 语义：程序语言中按语法规则构成的各个语法成分的含义，可分为静态语义和动态语义。

(3) 语用：表示构成语言的各个记号和使用者的关系，涉及符号的来源、使用和影响。

(4) 语境：理解和实现程序设计语言的环境，包括编译环境和运行环境。

4. 程序设计语言的发展

FORTRAN 是第一个被广泛用来进行科学计算的高级语言。

ALGOL60 是一个分程序结构的语言，它采用巴科斯范式(BNF)来描述语言的语法。

COBOL 是一种面向事务处理的高级语言，主要用于情报检索、商业数据处理等。

Pascal 是一种结构化程序设计语言，它从 ALGOL60 衍生而来。

C 语言是一种通用程序设计语言，允许程序员直接访问操作系统和底层硬件，因此在系统级应用和实时处理应用的开发中成为主要语言。

C++是在 C 语言的基础上发展起来的，增加了类机制，成为面向对象的程序设计语言。

Java 的初始用途是开发网络浏览器的小应用程序，目前已广泛应用于个人电脑(PC)、数据中心、游戏控制台、移动电话和互联网等。

PHP 是一种在服务器端执行的、嵌入 HTML 文档的脚本语言，可以快速地执行动态网页，其语法风格类似于 C 语言。

Python 是一种面向对象的解释型程序设计语言，可用于编写独立程序、快速脚本和复杂应用的原型。

Delphi 是一种可视化开发工具，主要特性是基于窗体和面向对象的方法、高速的编译器、强大的数据库支持、与 Windows 编程紧密结合以及成熟的组件技术。

5. 程序设计语言的分类

1)　命令式程序设计语言

命令式程序设计语言是基于动作的语言，在这种语言中，计算被看作动作的序列。命令式语言族开始于 FORTRAN、Pascal 和 C 语言，体现了命令式程序设计的关键思想。

2)　面向对象的程序设计语言

C++、Java 和 Smalltalk 是面向对象程序设计语言的代表。

3)　函数式程序设计语言

函数式程序设计语言是一类以λ-演算为基础的语言。该语言的代表是 LISP 语言，其中大量使用了递归。

4)　逻辑型程序设计语言

逻辑型程序设计语言是一类以形式逻辑为基础的语言。该语言的代表是建立在关系理论和一阶谓词理论基础上的 Prolog 语言。

真题链接

【例 2-1】可用于编写独立程序和快速脚本的语言是 (20) 。(2012 年 11 月真题 20)

(20) A. Python　　B. Prolog　　C. Java　　D. C#

【解析】Python 是一种面向对象的解释型程序设计语言，可用于编写独立程序、快速脚本和复杂应用的原型。Python 也是一种脚本语言，它支持对操作系统底层的访问。

Prolog 是一种逻辑型语言。Prolog 程序是一系列事实、数据对象或事实间的具体关系和规则的集合。Prolog 有很强的推理功能，适用于书写自动定理证明、专家系统、自然语言理解等问题的程序。

Java 是一种面向对象的程序设计语言，能开发应用在 Internet 上且具有软、硬件独立性和交互能力的程序。Java 可以一次编写而到处运行。

C#是微软公司发布的一种面向对象的、运行于.NET Framework(框架)之上的高级程序设计语言。C#看起来与 Java 有着惊人的相似：它包括了诸如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C#与 Java 有着明显的不同，它借鉴了 Delphi 的一个特点，与 COM(组件对象模型)是直接集成的，而且它是微软公司.NET Windows 网络框架的主角。

【答案】A

2.1.2 程序设计语言的基本成分

1. 数据成分

程序语言的数据成分是指一种程序语言的数据类型。

1) 常量和变量

按照程序运行时数据的值能否改变，将数据分为常量和变量。程序中的数据对象可以具有左值和(或)右值，左值是指存储单元(或地址、容器)，右值是指具体值(或内容)。变量具有左值和右值，在程序运行过程中其右值可以改变；常量只有右值，在程序运行过程中其右值不能改变。

2) 全局量和局部量

按数据的作用域范围，数据可分为全局量和局部量。系统为全局变量分配的存储空间在程序运行的过程中一般是不改变的，而为局部变量分配的存储单元是动态改变的。

3) 数据类型

按照数据组织形式的不同可将数据分为基本类型、用户定义类型、构造类型及其他类型。C(C++)的数据类型如下所示。

- 基本类型：整型(int)、字符型(char)、实型(float、double)和布尔型(bool)。
- 特殊类型：空类型(void)。
- 用户定义类型：枚举类型(enum)。
- 构造类型：数组、结构和联合。
- 指针类型：type *。
- 抽象数据类型：类类型。

其中，布尔类型和类类型是 C++在 C 语言的基础上扩充的。

2. 运算成分

程序语言的运算成分是指允许使用的运算符号及运算规则。大多数高级程序语言的基本运算可以分成算术运算、关系运算和逻辑运算，有些语言还提供位运算。运算符号的使用与数据类型密切相关。为了确保运算结果的唯一性，运算符号要规定优先级和结合性，必要时还要使用圆括号。

3. 控制成分

控制成分用于指明语言允许表述的控制结构，程序员使用控制成分来构造程序中的控制逻辑。

1) 顺序结构

在顺序结构中，计算过程从所描述的第一个操作开始，按顺序依次执行后续的操作，直到执行完序列的最后一个操作。顺序结构内也可以包含其他控制结构。

2) 选择结构

选择结构提供了在两种或多种分支中选择执行其中一个分支的逻辑。基本的选择结构是指定一个条件 P，然后根据条件的成立与否决定控制流走计算 A 还是走计算 B，从两个

分支中选择一个执行。

3)　循环结构

循环结构描述了重复计算的过程，通常包括三个部分：初始化、需要重复计算的部分和重复的条件。其中初始化部分有时在控制的逻辑结构中不进行显式的表示。循环结构主要有两种形式：while 型重复结构和 do-while 型重复结构。

4. C(C++)语言提供的控制语句

C(C++)语言提供的控制语句如下。

(1)　复合语句。复合语句是一系列用“{”和“}”括起来的声明和语句，其主要作用是将多条语句组成一个可执行单元。复合语句是一个整体。

(2)　if 语句和 switch 语句。这两种语句用于实现选择结构。

- if 语句实现的是双分支的选择结构。
- switch 语句描述了多分支的选择结构。

(3)　循环语句。C(C++)语言提供了三种形式的循环语句用于描述循环计算的控制结构。

- while 语句。
- do-while 语句。
- for 语句。

5. 函数

函数是程序模块的主要成分，它是一段具有独立功能的程序。函数的使用涉及三个概念：函数定义、函数声明和函数调用。

(1)　函数定义：包括函数首部和函数体两个部分。函数的定义描述了函数做什么和怎么做。

(2)　函数声明：函数应该先声明后引用。函数声明定义了函数原型。声明函数原型的目的在于告诉编译器传递给函数的参数个数、类型以及函数返回值的类型，参数表中仅需要依次列出函数定义中的参数的类型。编译器可以通过函数原型检查源程序中对函数的调用是否正确。

(3)　函数调用：当需要在一个函数(称为主调函数)中使用另一个函数(称为被调函数)实现的功能时，便以函数名字进行调用，称为函数调用。调用函数和被调用函数之间交换信息的方法主要有两种：一种是由被调用函数把返回值返回给主调函数，另一种是通过参数带回信息。函数调用时实参和形参之间交换信息的方法有传值调用和引用调用两种。

- 传值调用。若实现函数调用时实参向形式参数传递相应类型的值(副本)，则称为传值调用。这种方式下形式参数不能向实际参数传递信息。在 C 语言中，要实现被调用函数对实际参数的修改，必须用指针作形参。即调用时需要先对实参进行取地址运算，然后将实参的地址传递给指针形参，本质上仍属于传值调用。这种方式实现了间接内存访问。
- 引用调用。引用是 C++中增加的数据类型，当形式参数为引用类型时，形参名实际上是实参的别名，函数中对形参的访问和修改实际上就是针对相应实际参数所做的访问和改变。

真题链接

【例 2-2】弱类型语言(动态类型语言)是指不需要进行变量/对象类型声明的语言。__(50)__属于弱类型语言。(2013 年 11 月真题 50)

(50) A．Java　　B．C/C++　　C．Python　　D．C#

【解析】根据定义可知 Java、C/C++、C#都需要进行变量/对象声明，所以说是强类型语言。而 Python 不需要，它属于弱类型语言。

【答案】C

【例 2-3】程序运行过程中常使用参数在函数(过程)间传递信息，引用调用传递的是实参的__(20)__。(2014 年 5 月真题 21)

(20) A．地址　　B．类型　　C．名称　　D．值

【解析】引用就是某一变量(目标)的一个别名，对引用的操作与对变量直接操作完全一样。表示该引用名是目标变量名的一个别名。

【答案】C

2.2　语言处理程序基础

考核说明：本节主要介绍汇编语言、编译程序和解释程序的基本原理，编译程序部分经常考到，是重点也是难点，要能透彻理解。

2.2.1　汇编语言的基本原理

1. 汇编语言

汇编语言是为特定的计算机或计算机系统设计的面向机器的符号化的程序设计语言。用汇编语言编写的程序称为汇编语言源程序。

汇编语言源程序由若干条语句组成。一个程序中可以有三类语句：指令语句、伪指令语句和宏指令语句。

(1)　指令语句：又称为机器指令语句，汇编后能产生相应的机器代码，被 CPU 直接识别并执行相应的操作。指令语句可分为传送指令、算术运算指令、逻辑运算指令、移位指令、转移指令和处理机控制指令等。

(2)　伪指令语句：指示汇编程序在对源程序进行汇编时完成某些工作。与指令语句的区别是：伪指令语句经汇编后不产生机器代码，另外，伪指令语句所指示的操作是在源程序被汇编时完成的，而指令语句的操作必须在程序运行时完成。

(3)　宏指令语句：将多次重复使用的程序段定义为宏。宏的定义必须按照相应的规定进行，每个宏都有相应的宏名。

2. 汇编程序

汇编程序的功能是将用汇编语言编写的源程序翻译成机器指令程序。它一般至少需要

两次扫描源程序才能完成翻译过程。第一次扫描的主要工作是定义符号的值并创建一个符号表(ST)；第二次扫描的任务是产生目标程序。除了使用前一次扫描所产生的符号表(ST)外，还要使用机器指令表(MOT2)。在第二次扫描过程中，可执行汇编语句应被翻译成对应的二进制代码机器指令。这一过程涉及两个方面的工作：一是把机器指令助记符转换成二进制机器指令操作码，这可通过查找 MOT2 来实现；二是求出操作数区各操作的值(用二进制表示)。

2.2.2　编译程序的基本原理

编译程序的功能是把用高级语言书写的源程序翻译成与之等价的目标程序。编译过程划分成词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成六个阶段，实际的编译器可能会将其中的某些阶段结合在一起进行处理。

1. 词法分析阶段

词法分析阶段的任务是对源程序从前到后(从左到右)逐个字符进行扫描，从中识别出一个个“单词”符号。“单词”符号是程序设计语言的基本语法单位，如关键字、标识符等。词法分析程序输出的“单词”常常采用二元组的方式，即单词类别和单词自身的值。

2. 语法分析阶段

语法分析的任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位，如“表达式”、“语句”和“程序”等。词法分析和语法分析本质上都是对源程序的结构进行分析。

过程(函数)说明和过程(函数)调用是程序中一种常见的语法结构。过程说明和调用语句的翻译，有赖于形式参数和实际参数结合的方式以及数据空间的分配方式。需要分配存储空间的对象有基本数据类型、结构化数据类型和连接数据(如返回地址、参数等)。分配的依据是名字的作用域和生存期的定义规则。分配的策略有静态存储分配和动态存储分配两大类。

- 如果在编译时就能确定目标程序运行时所需要的全部空间大小，则在编译时就可以安排好目标程序运行时的全部数据空间，并确定每个数据对象的存储位置。这种分配策略为静态存储分配。
- 如果一个程序语言允许递归过程和可变数据结构，那么就需要采用动态存储分配技术。动态存储分配策略的实现有栈分配和堆分配两种方式。

3. 语义分析阶段

语义分析阶段主要是审查源程序是否存在语义错误，并收集类型信息供后面的代码生成阶段使用，只有语法和语义都正确的源程序才能翻译成正确的目标代码。语义分析的一个主要工作是进行类型分析和检查。

描述程序语义的形式化方法主要有属性文法、公理语义、操作语义和指称语义等，其中属性文法是对上下文无关文法的扩充。目前广泛使用的静态语义分析方法是语法制导翻译，其基本思想是将语言结构的语义以属性的形式赋予代表此结构的文法符号，而属性值

的计算以语义规则的形式赋予文法的产生式。在语法分析的推导或归纳的步骤中，通过语义规则实现对属性值的计算，以达到对语义的处理。

4. 中间代码生成阶段

中间代码是一种结构简单且含义明确的记号系统，可以有多种形式。中间代码生成阶段的工作就是根据语义分析的输出生成中间代码。语义分析和中间代码生成所依据的是语言的语义规则。

实际上，中间代码起着编译器前端和后端分水岭的作用，使用中间代码有利于提高编译程序的可移植性。常用的中间代码有后缀式、三元式、四元式和树等形状。

- 后缀式：把运算符写在运算对象的后面。例如，a*b 的后缀式为 ab*。这种表示法的优点是根据运算对象和运算符的出现次序进行计算，不需要使用括号，也便于用栈实现求值。
- 三元式：由运算符 OP、第一运算对象 ARG1 和第二运算对象 ARG2 组成。例如，x:=a+b 的三元式为：① (+, a, b) ② (:=, ①, x)。
- 四元式：组成成分为运算符 OP、第一运算对象 ARG1、第二运算对象 ARG2 和运算结果 RESULT。例如，x:=a+b 的四元式为：① (+, a, b, t) ② (:=, t, _, x)。
- 树：例如，表达式 x:=(a+b)*(c−d)可以表示为图 2-1 所示的结构。

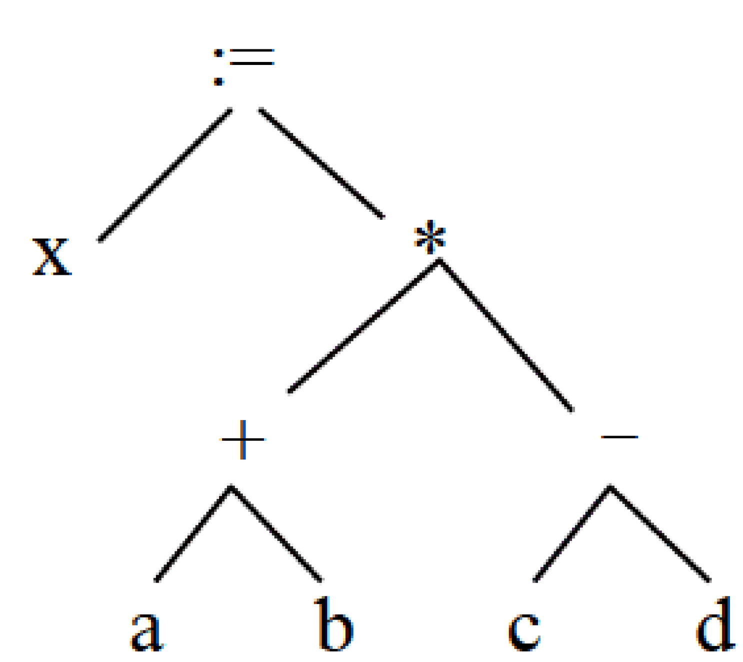


图 2-1　树型结构

5. 代码优化阶段

代码优化阶段的任务是对前阶段产生的中间代码进行变换或进行改造，目的是使生成的目标代码更为高效，即省时间和省空间。优化过程可以在中间代码生成阶段进行，也可以在目标代码生成阶段进行。

6. 目标代码生成阶段

目标代码生成阶段的任务是把中间代码变换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码。这是编译的最后阶段，它的工作与具体的机器密切相关。

7. 符号表管理

符号表的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成。符号表的建立可以始于词法分析阶段，也可以放到语法分析阶段，但符号表的使用有时会延续到目标代码的运行阶段。

8. 出错处理

用户编写的源程序中的错误大致可分为静态错误和动态错误。动态错误也称动态语义错误，指程序中包含的逻辑错误。静态错误是指编译阶段发现的程序错误，可分为语法错误和静态语义错误。出错处理程序的任务包括检查错误、报告出错信息、排错、恢复编译工作。

真题链接

【例 2-4】将高级语言源程序翻译成机器语言程序的过程，常引入中间代码。以下关于中间代码的叙述中，不正确的是__(22)__。(2014 年 11 月真题 22)

(22) A．中间代码不依赖于具体的机器

B．使用中间代码可提高编译程序的可移植性

C．中间代码可以用树或图表示

D．中间代码可以用栈和队列表示

【解析】中间代码是源程序的一种内部表示，或称中间语言。中间代码的作用是可使编译程序的结构在逻辑上更为简单明确，使用中间代码可提高编译程序的可移植性，常见的有逆波兰记号、四元式、三元式和树。

【答案】D

【例 2-5】对高级语言源程序进行编译的过程可以分为多个阶段，分配寄存器的工作在__(48)__阶段进行。(2014 年 11 月真题 48)

(48) A．词法分析　　B．语法分析

C．语义分析　　D．目标代码生成

【解析】目标代码生成阶段应考虑直接影响到目标代码速度的三个问题：一是如何生成较短的目标代码；二是如何充分利用计算机中的寄存器，减少目标代码访问存储单元的次数；三是如何充分利用计算机指令系统的特点，以提高目标代码的质量。

【答案】D

【例 2-6】编译程序对高级语言源程序进行编译的过程中，要不断收集、记录和使用源程序中一些相关符号的类型和特征等信息，并将其存入__(22)__中。 (2014 年 5 月真题 22)

(22) A．符号表　　B．哈希表　　C．动态查找表　　D．栈和队列

【解析】编译过程中编译程序不断汇集和反复查证出现在源程序中各种名字的属性和特征信息等有关信息。这些信息通常记录在一张或几张符号表中。符号表的每一项都有两个部分：一部分是名字(标识符)；一部分是名字属性(标识符的有关信息)。编译过程中，每当扫描器(词法分析器)识别出一个名字后，编译程序就查阅符号表，看其是否在符号表中。符号表在编译全过程的地位和作用非常重要，是进行上下文合法性检查和语义处理及代码生成的依据。符号表总体结构的设计和实现既与源语言的复杂性(包括词法结构、语法结构的复杂性)有关，还与对于编译系统在时间效率和空间效率方面的要求有关。

【答案】A

【例 2-7】在某 C/C++程序中，整型变量 a 的值为 0 且应用在表达式“c=b/a”中，则最可能发生的情形是__(50)__。(2014 年 5 月真题 50)

(50) A．编译时报告有语法错误　　　　　　B．编译时报告有逻辑错误

C．运行时报告有语法错误　　　　　　D．运行时产生异常

【解析】编译时 a 的值无法确定，表达式“c=b/a”符合 C/C++语言的语法逻辑，编译时不会报错。运行时，代入 a 的值，发生错误。

【答案】D

2.2.3　解释程序的基本原理

解释程序是一种语言处理程序，在词法、语法和语义分析方面与编译程序的工作原理基本相同，但在运行用户程序时，它直接执行源程序或源程序的内部形式(中间代码)。因此，解释程序并不产生目标程序，这是它和编译程序的主要区别。

解释程序的结构通常可以分成两部分：第一部分是分析部分，包括通常的词法分析、语法分析和语义分析程序，经语义分析后把源程序翻译成中间代码，中间代码常采用逆波兰表示形式；第二部分是解释部分，用来对第一部分产生的中间代码进行解释执行。

真题链接

【例 2-8】以下关于实现高级程序设计语言的编译和解释方式的叙述中，正确的是 (48) 。(2014 年 5 月真题 48)

(48) A．在编译方式下产生源程序的目标程序，在解释方式下不产生

B．在解释方式下产生源程序的目标程序，在编译方式下不产生

C．编译和解释方式都产生源程序的目标程序，差别是优化效率不同

D．编译和解释方式都不产生源程序的目标程序，差别在是否优化

【解析】在编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序要参与到程序的运行过程中，运行程序的控制权在解释程序。解释器在翻译源程序时不产生独立的目标程序，而编译器则需要将源程序翻译成独立的目标程序。

【答案】A

2.3　文法和有限自动机

考核说明：本节主要介绍文法和语言的形式描述、词法分析、正规式和有限自动机之间的转换、词法分析器的构造和语法分析。其中，正规式和有限自动机是考核重点，也是难点，要正确理解并掌握。

2.3.1　文法和语言的形式描述

1. 字符串的相关表示

字母表：字符的非空有穷集合，记为 Σ。

空串：由 0 个字符组成的序列，记为ε。

连接：字符串 S 和字符串 T 的连接是指将 T 接续在 S 之后，记为 $S{\cdot}T$。

Σ^*：包括空串在内的 Σ 上的所有字符串的集合。

字符串的方幂：把字符串 α 自身连接 n 次得到的串，记为 α^n。

正则闭包+：$A^+=A^1\cup A^2\cup A^3\cup\cdots\cup A^n\cup\cdots$。

闭包*：$A^*=A^0\cup A^+$。

2. 文法的定义

描述语言语法结构的形式规则称为文法。文法 G 是一个四元组，可表示为 $G=(V_N, V_T, P, S)$，其中 V_T 是一个非空有限集，其中的每个元素称为一个终结符；V_N 是一个非空有限集，其每个元素称为非终结符。$V_N\cap V_T=\varnothing$。令 $V=V_N\cup V_T$，称 V 为文法 G 的词汇表。P 是产生式的有限集合，每个产生式是形如 $\alpha\rightarrow\beta$ 的规则，其中 α 称为产生式的左部，$\alpha\in V^+$且 α 中至少含有一个非终结符；β 称为产生式的右部，且 $\beta\in V^*$。$S\in V_N$，称为开始符号，它至少要在一条产生式中作为左部出现。

3. 文法的分类

乔姆斯基把文法分成四种类型，即 0 型、1 型、2 型和 3 型。

- 0 型文法：0 型文法也称为短语文法，其能力相当于图灵机。
- 1 型文法：G 的任何产生式 $\alpha\rightarrow\beta$ ($S\rightarrow\varepsilon$ 除外)均满足$|\alpha|\leqslant|\beta|$($|x|$表示 x 中文法符号的个数)。1 型文法也称为上下文有关文法，这种文法意味着对非终结符的替换必须考虑上下文，并且一般不允许替换成ε串，此文法对应于线性有界自动机。
- 2 型文法：G 的任何产生式 $A\rightarrow\beta$，其中 $A\in V_N$，$\beta\in V^*$。2 型文法也称为上下文无关文法，对非终结符的替换无须考虑上下文，它对应于下推自动机。
- 3 型文法：G 的任何产生式 $A\rightarrow a$ 或 $A\rightarrow aB$(或者 $A\rightarrow Ba$)，其中 $A,B\in V_N$，$a\in V_T$。3 型文法等价于正规式，因此也称为正规文法或线性文法，它对应于有限状态自动机。

真题链接

【例 2-9】对于大多数通用程序设计语言，用<u>(50)</u>描述其语法即可。(2014 年 11 月真题 50)

(50) A．正规文法　　B．上下文无关文法　　C．上下文有关文法　　D．短语结构文法

【解析】上下文无关文法：形式语言理论中一种重要的变换文法，用来描述上下文无关语言，在乔姆斯基分层中称为 2 型文法，由于程序设计语言的语法基本上都是上下文无关文法，因此应用十分广泛。

【答案】B

2.3.2　词法分析

1. 正规式和正规集

对于字母表 Σ，其上的正规式(也称正则表达式，正规表达式)及其表示的正规集可以递归定义如下。

(1) ε是一个正规式，它表示集合$L(\varepsilon)=\{\varepsilon\}$。

(2) 若 a 是 Σ 上的字符，则 a 是一个正规式，它所表示的正规集为$\{a\}$。

(3) 若正规式 r 和 s 分别表示正规集 $L(r)$和 $L(s)$，则

- $r|s$ 是正规式，表示集合 $L(r)\cup L(s)$。
- $r \cdot s$ 是正规式，表示集合 $L(r)L(s)$。
- r^*是正规式，表示集合$(L(r))^*$。
- (r)是正规式，表示集合 $L(r)$。

仅由有限次地使用上述三个步骤定义的表达式才是 Σ 上的正规式，其中运算符“|”、“·”、“*”分别称为“或”、“连接”和“闭包”。若两个正规式表示的正规集相同，则认为两者等价。

2. 有限自动机

有限自动机(也称有穷自动机)是一种识别装置的抽象概念，它能够正确地识别正规集。

1) 确定的有限自动机

一个确定的有限自动机(DFA)是个五元组(S, Σ, f, s_0, Z)，其中：

- S 是一个有限集，其每个元素称为一个状态。
- Σ 是一个有限字母表，其每个元素称为一个输入字符。
- f 是从 $S\times\Sigma\rightarrow S$ 上的单值部分映像。
- $s_0\in S$ 是唯一的一个开始状态。
- Z 是非空的终止状态集合。

一个 DFA 可以用两种直观的方式表示：状态转换图和状态转换矩阵。状态转换图简称为转换图，它是一个有向图。DFA 中的每个状态对应转换图中的一个结点，DFA 中的每个转换函数对应图中的一条有向弧，若转换函数为$f(A, a)=Q$，则该有向弧从结点 A 出发，进入结点 Q，字符 a 是弧上的标记。状态转换矩阵可以用一个二维数组 M 表示，矩阵元素的行下标表示状态，列下标表示输入字符，$M[A, a]$的值是当前状态为 A、输入为 a 时，应转换到的下一状态。在转换矩阵中，一般以第一行的行下标所对应的状态作为初态，而终态则需要特别指出。

2) 不确定的有限自动机

一个不确定的有限自动机(NFA)也是一个五元组，它与确定的有限自动机的区别如下。

- f 是从 $S\times\Sigma\rightarrow 2^S$ 上的映像。对于 S 中的一个给定状态及输入符号，返回一个状态的集合。
- 有向弧上的标记可以是ε。

显然，DFA 是 NFA 的特例。

实际上，对于每个 NFA M，都存在一个 DFA N，且 $L(M)=L(N)$。

对于任何两个有限自动机 M1 和 M2，如果 L(M1)=L(M2)，则称 M1 和 M2 是等价的。

3. NFA 到 DFA 的转换

设 NFA $N=(S, \Sigma, f, s_0, Z)$，与之等价的 DFA $M=(S', \Sigma, f', q_0, Z')$，用子集法将非确定的有限自动机确定化的算法步骤如下。

(1) 求出 DFA M 的初态 q_0，此时 S' 仅含初态 q_0，并且没有标记。

(2) 对于 S' 中尚未标记的状态 $q_i = \{s_{i1}, s_{i2}, \cdots, s_{im}\}$ 和 $s_{ij} \in S(j=1,2,\cdots,m)$ 进行下述处理。

① 标记 q_i。

② 对于每个 $a \in \Sigma$，令 $T = f(s_{i1}, s_{i2}, \cdots, s_{im}, a)$，$q_j = \varepsilon_\mathrm{CLOSURE}(T)$。

③ 若 q_i 尚不在 S' 中，则将 q_j 作为一个未加标记的新状态添加到 S'，并把状态转换函数 $f'(q_i, a) = q_j$ 添加到 DFA M。

(3) 重复步骤(2)，直到 S' 中不再有未标记的状态时为止。

(4) 令 $Z' = \{q | q \in S' 且 q \cap Z \neq \varnothing\}$。

注意：若 I 是 NFA N 的状态集合的一个子集，其中 $\varepsilon_\mathrm{CLOSURE}(I)$ 的定义如下。

① 状态集 I 的 $\varepsilon_\mathrm{CLOSURE}(I)$ 是一个状态集。

② 状态集 I 的所有状态属于 $\varepsilon_\mathrm{CLOSURE}(I)$。

③ 若 s 在 I 中，那么从 s 出发经过任意条 ε 弧到达的状态 s' 都属于 $\varepsilon_\mathrm{CLOSURE}(I)$。

从 NFA 转换得到的 DFA 不一定是最简化的，可以通过等价变换将 DFA 进行最小化处理。

真题链接

【例 2-10】以下关于图 2-2 所示有限自动机的叙述中，不正确的是 __(21)__。(2014 年 11 月真题 49)

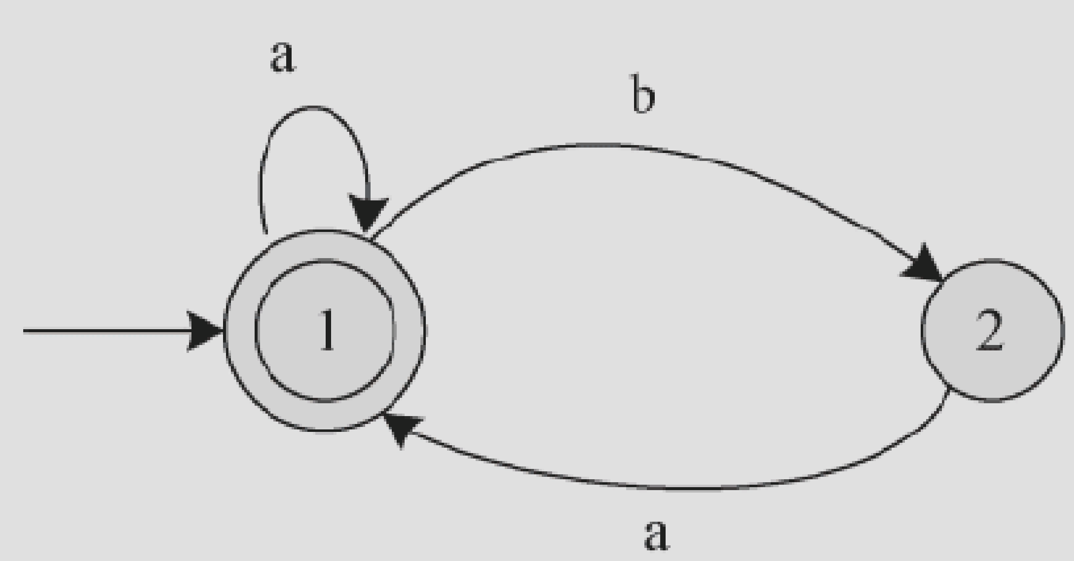


图 2-2　有限自动机的状态转换图

(21) A．该自动机识别的字符串中 a 不能连续出现　　B．自动机识别的字符串中 b 不能连续出现

C．自动机识别的非空字符串必须以 a 结尾　　D．自动机识别的字符串可以为空串

【解析】图中 a 可代表两个步骤：状态 1→状态 1，状态 2→状态 1。如果两个 a 连续出现，则无法区分。

【答案】A

【例 2-11】已知文法 $G: S \to A0 | B1, A \to S1 | 1, B \to S0 | 0$，其中 S 是开始符号。从 S 出发可以推导出 __(21)__。(2013 年 11 月真题 21)

(21) A．所有由 0 构成的字符串　　B．所有由 1 构成的字符串

C．某些 0 和 1 个数相等的字符串　　D．所有 0 和 1 个数不同的字符串

【解析】从开始符出发，能推导出两种串：一种以 0 结尾，一种以 1 结尾。以 0 结尾的前面必须是 1，而这个 1 前面可能还有一个递归的 S；以 1 结尾的前面必须是 0，而这个 0 前面可能还有一个递归的 S。由此可以知道，这是文法可以导出某些 0 和 1 的个数相同的串。之所以说是某些，而不是所有，是因为，该文法所导出的串中 0 附近必有 1，1 附近必有 0，比如：01,0101,0110 等等，但 000111 就不能导出。

【答案】C

2.3.3 正规式与有限自动机之间的转换

(1) 对于 Σ 上的 NFA M，可以构造一个 Σ 上的正规式 R，使得 $L(R)=L(M)$。

构造过程分两步进行。

① 在 M 的状态转换图中加两个结点 x 和 y。

② 按图 2-3 所示的方法逐步消去 M 中的除 x 和 y 的所有结点。

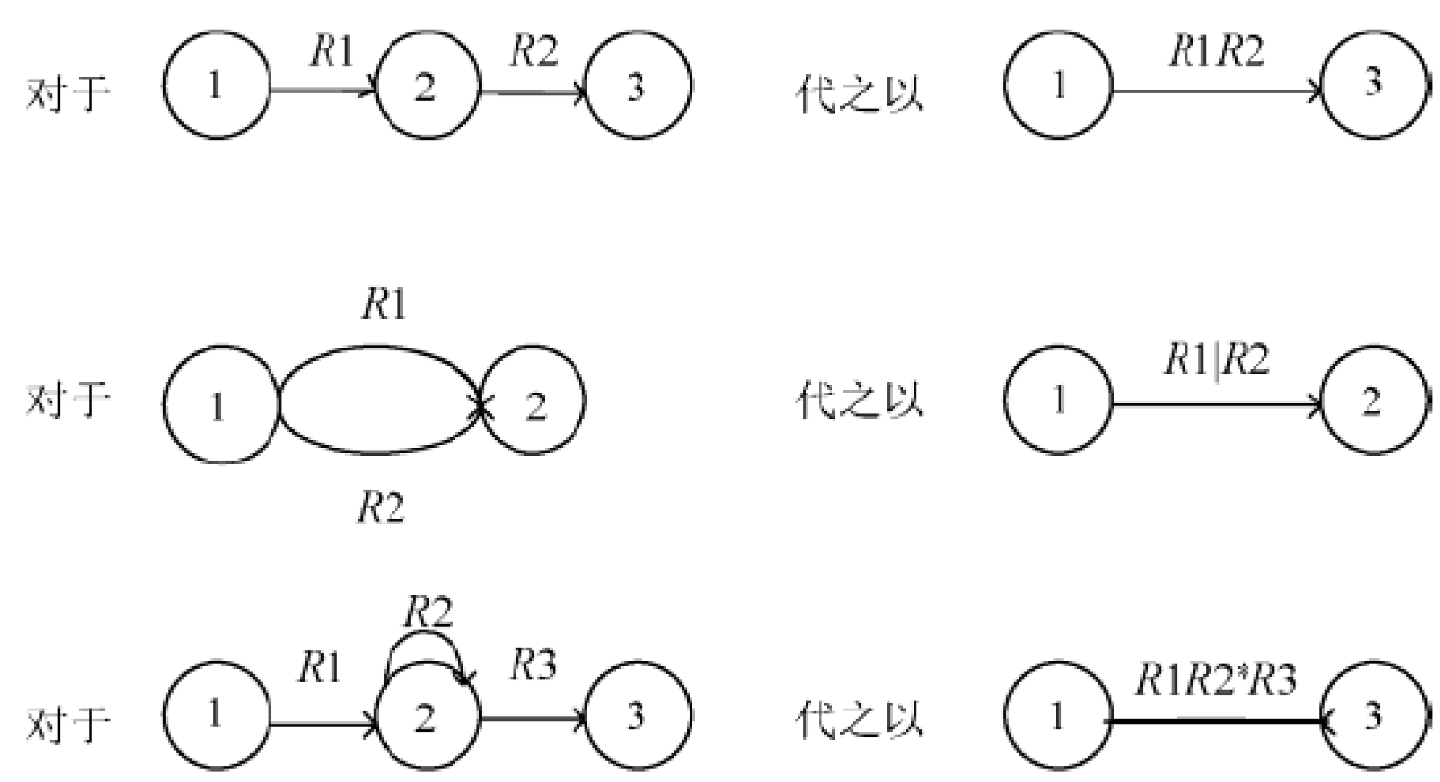


图 2-3 状态转换图(消去中间结点)

(2) 对于 Σ 上的每一个正规式 R，可以构造一个 Σ 上的 NFA M，使得 $L(M)=L(R)$。

构造过程分两步进行。

① 对于正规式 R，可用图 2-4 所示的拓广状态图表示。

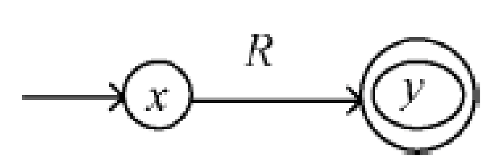


图 2-4 拓广状态图

② 通过对正规式 R 进行分裂并加入新的结点，逐步把图转变成每条弧上的标记是 Σ 上的一个字符或 ε，转换规则如图 2-5 所示。

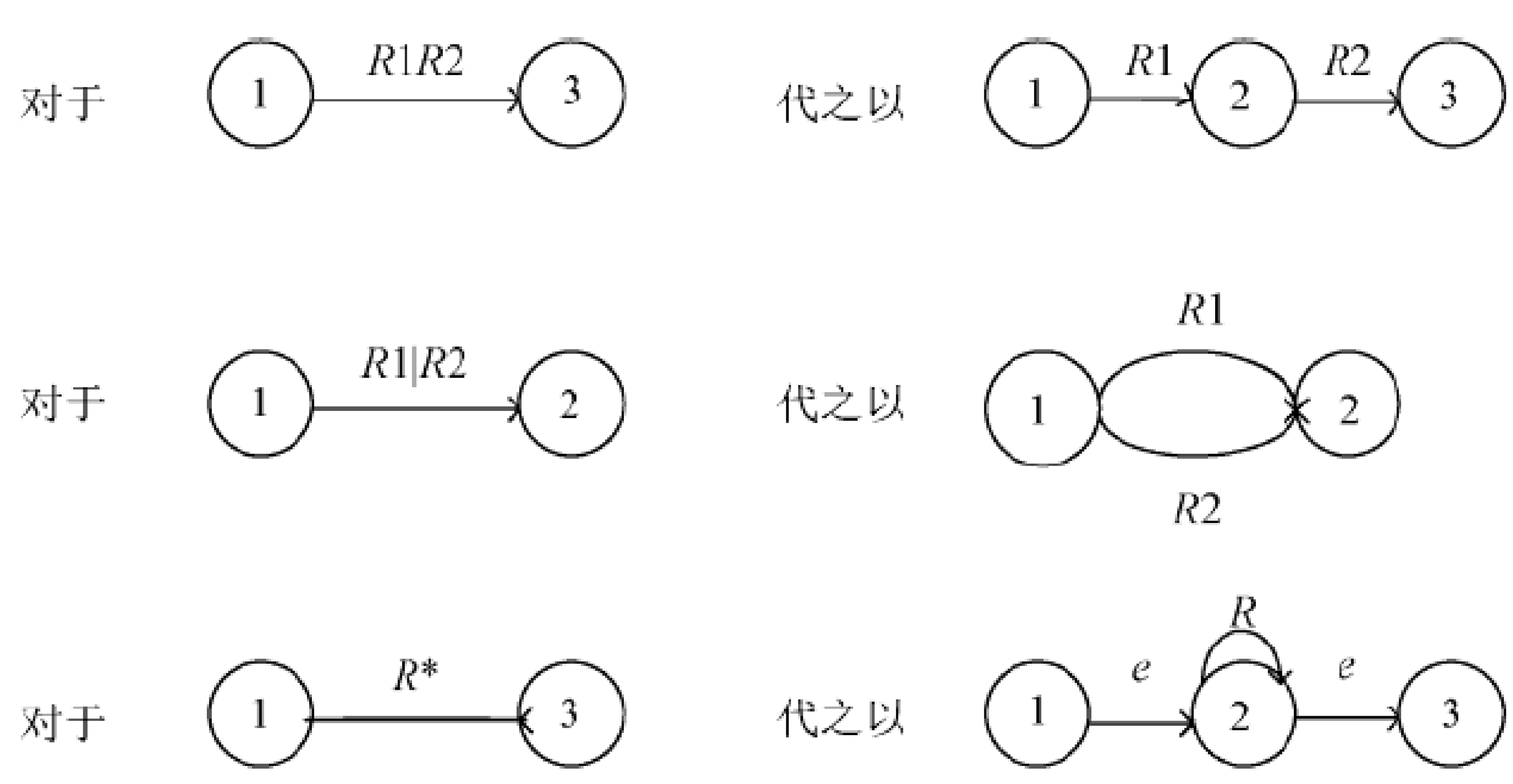


图 2-5 状态转换图(加入新结点)

真题链接

【例 2-12】图 2-6 所示为一个有限自动机(其中，A 是初态，C 是终态)，该自动机所识别的字符串的特点是__(48)__。(2012 年 5 月真题 48)

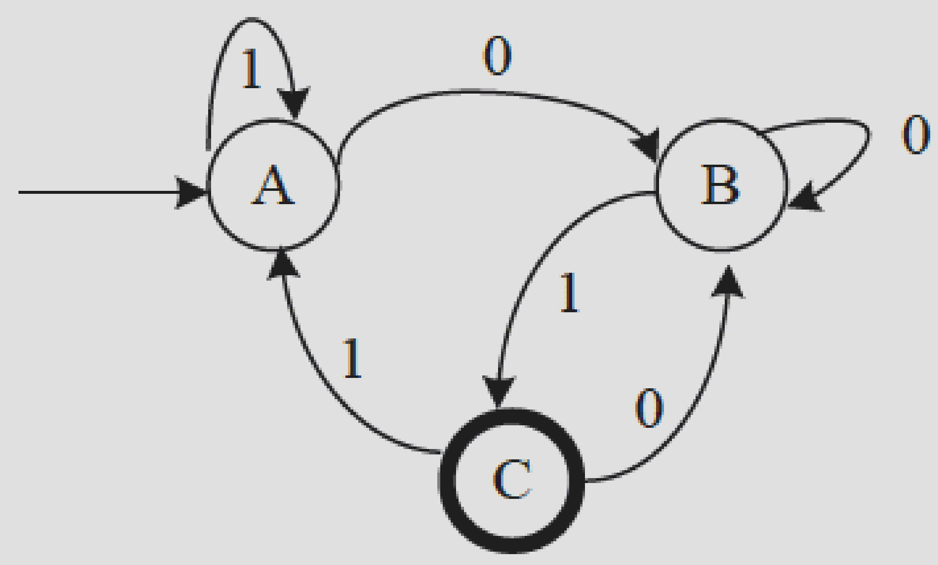


图 2-6　有限自动机的状态转换图

(48) A. 必须以 11 结尾的 0、1 串　　B. 必须以 00 结尾的 0、1 串

C. 必须以 01 结尾的 0、1 串　　D. 必须以 10 结尾的 0、1 串

【解析】有限自动机可识别的字符串，是指从有限自动机的初态出发，存在一条到达终态的路径，其上的标记所构成的字符串。本题 C 是终态，C 的前一状态只能是 B，由 B 到 C 输入的是 1；B 的前一状态可以是 A，也可以是 B，也可以是 C，但输入的都是 0。可见，该有限自动机识别的串必须以 01 结尾。

【答案】C

2.3.4　词法分析器的构造

词法分析器的构造过程如下。

(1) 用正规式描述语言中的单词构成规则。

(2) 为每个正规式构造一个 NFA，用于识别正规式所表示的正规集。

(3) 将构造出的 NFA 转换成等价的 DFA。

(4) 对 DFA 进行最小化处理，使其最简。

(5) 根据 DFA 构造词法分析器。

2.3.5　语法分析

语法分析的任务是根据语言的语法规则，分析单词串是否构成短语和句子，同时检查和处理程序中的语法错误。根据产生语法树的方向，语法分析可分为自底向上和自顶向下两类。

所谓自顶向下的分析是对给定的符号串，试图自顶向下地为其构造一棵语法树，或者说从文法的开始符号出发，为其构造一个最左推导。

所谓自底向上的分析是对给定的符号串，试图自底向上地为其构造一棵语法树，或者说从给定的符号串本身出发，试图将其归约为文法的开始符号。

算符优先文法属于自底向上的分析法，它利用各个算符间的优先关系和结合规则来进行语法分析，特别是用于分析各种表达式。算符优先文法的任何产生式的右部都会出现两个非终结符相邻的情况，且任何一对终结符之间至多只有 3 种算符关系“>”、“<”和“=”之一成立。

真题链接

【例 2-13】算术表达式“(a-b)*(c+d)”的后缀式是 (21) 。(2014 年 11 月真题 21)

(21) A. ab-cd+*　　B. abcd-*+　　C. ab-*cd+　　D. ab-c+d*

【解析】后缀式是波兰逻辑学家卢卡西维奇发明的一种表达方式，即把运算符号写在运算对象的后面，例如把 a+b 写成 ab+，这种表示法的优点是根据运算对象和算符的出现次序进行计算，不需要使用括号。

【答案】A

2.4 应试加油站

2.4.1 考频统计

通过对近几年考试真题的分析，“程序语言基础”知识模块，在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 2-1 所示。

表 2-1 历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：21～22、48～50	函数传递、解释和编译、文法和词法分析	5
	下午：无		0
2014 年 5 月	上午：21～22、48～50	文法分析、有限自动机、解释和编译、中间代码	5
	下午：无		0
2013 年 11 月	上午：20～22、48～50	函数调用、文法分析、后缀式、程序翻译、有限自动机、弱类型语言	6
	下午：无		0
2013 年 5 月	上午：20～22、48～50	解释程序和编译程序的概念、函数调用、编译程序、文法分析、编译程序	6
	下午：无		0
2012 年 11 月	上午：20～22、48～50	脚本语言、正规式、后缀式、编译程序、中间代码表示形式、程序错误	6
	下午：无		0
2012 年 5 月	上午：20、22、48、50～51	逻辑表达式、后缀式、有限自动机、参数传递、编译和解释的不同	5
	下午：无		0

纵观历年试卷，本章知识点是以选择题的形式出现在试卷中的。本章知识点在历次考试上午试卷中，所考查的题量为 5～6 道选择题，所占分值为 5～6 分(占试卷总分值 75 分中的 7%～8%)；在下午试卷中没有相关考题。本章考题主要检验考生是否理解相关的理论知识点，有限状态机和正规式是重点也是难点。

2.4.2 解题技巧

【典型题 2-1】在引用调用方式下进行函数调用是将__(21)__。(2013 年 5 月真题 21)

(21) A．实参的值传递给形参　　　B．实参的地址传递给形参

C．形参的值传递给实参　　　D．形参的地址传递给实参

【解析】引用调用是把实参(如 int a)的地址(&a)赋给形参(指针变量，比如*b，这时 b=&a，即 b 指向变量 a)，如果*b(也即 a 对应的内存空间)发生变化，也就是变量 a 的值发生了变化。

【答案】B

【典型题 2-2】程序运行过程中常使用参数在函数(过程)间传递信息，引用调用传递的是实参的__(20)__。(2013 年 11 月真题 20)

(20) A．地址　　　B．类型　　　C．名称　　　D．值

【解析】引用调用把参数的地址复制给形式参数。在函数内，该地址用于访问调用中要用到的实际参数。这意味着，修改形式参数会影响实际参数。

【答案】A

【典型题 2-3】可用于编写独立程序和快速脚本的语言是__(20)__。(2012 年 11 月真题 20)

(20) A．Python　　　B．Prolog　　　C．Java　　　D．C#

【解析】Python 是一种面向对象的解释型程序设计语言，可用于编写独立程序、快速脚本和复杂应用的原型。Python 也是一种脚本语言，它支持对操作系统底层的访问。

Prolog 是一种逻辑型语言。Prolog 程序是一系列事实、数据对象或事实间的具体关系和规则的集合。Prolog 有很强的推理功能，适用于书写自动定理证明、专家系统、自然语言理解等问题的程序。

Java 是一种面向对象的程序设计语言，能开发应用在 Internet 上且具有软、硬件独立性和交互能力的程序。Java 可以一次编写而到处运行。

C#是微软公司发布的一种面向对象的、运行于.NET Framework(框架)之上的高级程序设计语言。C#看起来与 Java 有着惊人的相似：它包括了诸如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C#与 Java 有着明显的不同，它借鉴了 Delphi 的一个特点，与 COM(组件对象模型)是直接集成的，而且它是微软公司.NET Windows 网络框架的主角。

【答案】A

【典型题 2-4】对于逻辑表达式“x and y or not z”，and、or、not 分别是逻辑与、或、非运算，优先级从高到低为 not、and、or，and、or 为左结合，not 为右结合，若进行短路计算，则__(20)__。(2012 年 5 月真题 20)

(20) A. x 为真时，整个表达式的值为真，不需要计算 y 和 z 的值

B. x 为假时，整个表达式的值为假，不需要计算 y 和 z 的值

C. x 为真时，根据 y 的值决定是否需要计算 z 的值

D. x 为假时，根据 y 的值决定是否需要计算 z 的值

【解析】“x and y or not z”可以表示为“(x and y) or (not z)”。x 为真时，如果 y 为真，则 x and y 的结果为真，此时不需要计算 z 的值，整个表达式的值为真；如果 y 为假，则 x and y 的结果为假，此时整个表达式的值由 z 决定，如果 z 为真，则 not z 为假，最终结果为假，反则为真。

x 为假时，不需要计算 y 的值，x and y 的结果为假，整个表达式的值由 z 决定。

【答案】C

【典型题 2-5】函数(过程)调用时，常采用传值与传地址两种方式在实参和形参间传递信息。以下叙述中，正确的是__(50)__。(2012 年 5 月真题 50)

(50) A. 在传值方式下，将形参的值传给实参，因此，形参必须是常量或变量

B. 在传值方式下，将实参的值传给形参，因此，实参必须是常量或变量

C. 在传地址方式下，将形参的值传给实参，因此，形参必须有地址

D. 在传地址方式下，将实参的值传给形参，因此，实参必须有地址

【解析】传值调用是指传递变量值的调用方式。在这种调用方式中，实参使用变量名或者表达式，形参使用变量名。在调用时，调用函数将实参值拷贝一个副本给形参，使形参按顺序从对应的实参中获得值，这就相当于将实参值对应地赋给形参，使形参获值。传地址调用是指在调用时传递变量地址值的传值调用。传地址调用时要求调用函数的实参用地址值，而被调用函数的形参用指针，于是函数之间进行地址值的传递。这种传递是将实参的变量地址值传递给形参指针，即让形参指针指向实参变量，这种传递方式与调用函数拷贝实参值的副本给形参是不同的，它是让形参指针直接指向实参的变量。

【答案】D

【典型题 2-6】以下关于解释程序和编译程序的叙述中，正确的是__(20)__。(2013 年 5 月真题 20)

(20) A. 编译程序和解释程序都生成源程序的目标程序

B. 编译程序和解释程序都不生成源程序的目标程序

C. 编译程序生成源程序的目标程序，解释程序则不然

D. 编译程序不生成源程序的目标程序，而解释程序反之

【解析】编译程序，也称编译器，用于将源程序翻译成目标语言程序，然后在计算机上运行目标程序。虽然执行效率高，但编写出来的程序可读性很差，且难以修改和维护。

解释程序，也称解释器，它或者直接解释执行源程序，或者将源程序翻译成某种中间代码后再加以执行。

【答案】C

【典型题 2-7】在对高级语言源程序进行编译的过程中，为源程序中变量所分配的存储单元的地址属于__(22)__。(2013 年 5 月真题 22)

(22) A. 逻辑地址　　B. 物理地址　　C. 接口地址　　D. 线性地址

【解析】在编译时产生的目标文件都是从地址 0 开始的，在连接时将各个目标文件进行符号替换，这时会修改相应的地址，最后产生一个从地址 0 开始的可执行文件。在该可执行文件中的地址称为逻辑地址。

【答案】A

【典型题 2-8】编译过程中，对高级语言程序语句的翻译主要考虑声明语句和可执行语句。对声明语句，主要是将需要的信息正确地填入合理组织的__(49)__中；对可执行语句，则是__(50)__。(2013 年 5 月真题 49～50)

(49) A. 符号表　　B. 栈　　C. 队列　　D. 树

(50) A. 翻译成机器代码并加以执行　　B. 转换成语法树

C. 翻译成中间代码或目标代码　　D. 转换成有限自动机

【解析】声明语句的作用是为可执行语句提供信息，以便于其执行。对声明语句的处理，主要是将所

需要的信息正确地填写进合理组织的符号表中。符号表的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效的查找、插入、修改和删除等操作。经过词法、语法、语义分析之后可执行语句被翻译成中间代码或目标代码。

【答案】(49)A；(50)C

【典型题 2-9】将高级语言程序翻译为机器语言程序的过程中，常引入中间代码，其好处是__(48)__的适当工具。(2013 年 11 月真题 48)

(48) A．有利于进行反编译处理　　B．有利于进行与机器无关的优化处理

C．尽早发现语法错误　　D．可以简化语法和语义分析

【解析】“中间代码”是一种简单且含义明确的记号系统，与具体的机器无关，可以有若干种形式。可以将不同的高级程序语言翻译成同一种中间代码。由于与具体机器无关，使用中间代码有利于进行与机器无关的优化处理，以及提高编译程序的可移植性。

【答案】B

【典型题 2-10】以下关于程序错误的叙述中，正确的是__(50)__。(2012 年 11 月真题 50)

(50) A．编译正确的程序必然不包含语法错误

B．编译正确的程序必然不包含语义错误

C．除数为 0 的错误可以在语义分析阶段检查出来

D．除数为 0 的错误可以在语法分析阶段检查出来

【解析】用户编写的源程序不可避免地会有一些错误，这些错误大致可以分为静态错误和动态错误。动态错误也称动态语义错误，它们发生在程序运行时，例如除数为 0、引用数组元素下标错误等。静态错误是指编译阶段发现的程序错误，可分为语法错误和静态语义错误，如单词拼写错误、标点符号错、表达式缺少操作数、括号不匹配等有关语言结构上的错误称为语法错误，而语义分析时发现的运算符与运算对象不合法等错误属于静态语义错误。

【答案】C

【典型题 2-11】在对程序语言进行翻译的过程中，常采用一些与之等价的中间代码表示形式。常用的中间代码表示不包括__(49)__。(2012 年 11 月真题 49)

(49) A．树　　B．后缀式　　C．四元式　　D．正则式

【解析】常用的中间代码有后缀式、三元式、四元式和树等形式。

【答案】D

【典型题 2-12】编译和解释是实现高级程序设计语言翻译的两种基本形式。以下关于编译与解释的叙述中，正确的是__(51)__。(2012 年 5 月真题 51)

(51) A. 在解释方式下，对源程序不进行词法分析和语法分析，直接进行语义分析

B. 在解释方式下，无须进行词法、语法和语义分析，而是直接产生源程序的目标代码

C. 在编译方式下，必须进行词法、语法和语义分析，然后产生源程序的目标代码

D. 在编译方式下，必须先形成源程序的中间代码，然后再产生与机器对应的目标代码

【解析】编译程序是将高级语言书写的源程序翻译成与之等价的低级语言的目标程序，编译的工作阶段包括：词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成。解释程序直接执行源程序或源程序的内部形式，一般是读一句源程序，翻译一句，执行一句，不产生目标代码，如 BASIC 解释程序。编译程序与解释程序最大的区别之一在于前者生成目标代码，而后者不生成；此外，前者产生的

目标代码的执行速度比解释程序的执行速度要快；后者人机交互好，适于初学者使用。

【答案】C

【典型题 2-13】大多数程序设计语言的语法规则用__(49)__描述即可。(2014 年 5 月真题 49)

(49) A．正规文法　　B．上下文无关文法　　C．上下文有关文法　　D．短语结构文法

【解析】上下文无关文法是形式语言理论中一种重要的变换文法，用来描述上下文无关语言，在乔姆斯基分层中称为 2 型文法。由于程序设计语言的语法基本上都是上下文无关文法，因此应用十分广泛。上下文无关文法拥有足够强的表达力来表示大多数程序设计语言的语法。另一方面，上下文无关文法又足够简单，使得我们可以构造有效的分析算法来检验一个给定字符串是否是由某个上下文无关文法产生的。

【答案】B

【典型题 2-14】以下关于语言 $L=\{a^nb^n|n>1\}$ 的叙述中，正确的是__(48)__。(2013 年 5 月真题 48)

(48) A．可用正规式“aa*bb*”描述，但不能通过有限自动机识别

B．可用正规式“a^mb^m”表示，也可用有限自动机识别

C．不能用正规式表示，但可以用有限自动机识别

D．不能用正规式表示，也不能通过有限自动机识别

【解析】根据正规式和有限自动机的规范，结合题中给出的语言 L，很明显都无法将语言 L 表示和识别出来。

【答案】D

【典型题 2-15】算术表达式 a+(b−c)*d 的后缀式是__(22)__。(−、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例)(2013 年 11 月真题 22)

(22) A．bc−d*a+　　B．abc−d*+　　C．ab+c−d*　　D．abcd−*+

【解析】后缀表达式：不包含括号，运算符放在两个运算对象的后面，所有的计算按运算符出现的顺序，严格从左向右进行。

【答案】B

【典型题 2-16】对高级语言源程序进行编译的过程中，有穷自动机(NFA 或 DFA)是进行__(49)__。(2013 年 11 月真题 49)

(49) A．词法分析　　B．语法分析　　C．语义分析　　D．出错处理

【解析】DFA 对于同一个输入只有一个确定的下一状态，所以词法分析器优先采用它。

【答案】A

【典型题 2-17】语言 $L=\{a^mb^n|m\geq0, n\geq1\}$ 的正规表达式是__(21)__。(2012 年 11 月真题 21)

(21) A．aa^*bb^*　　B．a^*bb^*　　C．aa^*b^*　　D．a^*b^*

【解析】$m\geq0$，a^m 说明可以有 0 个 a 或者多个 a；$n\geq1$，b^n 说明至少有 1 个 b 或者多个 b。a^*表示由 0 个或者多个 a 构成的集合，可以表示$\{a^m|m\geq0\}$；b^*表示由 0 个或者多个 b 构成的集合，bb^*才可以表示$\{b^n|n\geq1\}$。

【答案】B

【典型题 2-18】算术表达式(a−b)*c+d 的后缀式是__(22)__(−、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例)。(2012 年 11 月真题 22)

(22) A．abcd− * +　　B．ab−cd*+　　C．ab−c*d+　　D．abc−d*+

【解析】后缀表达式的计算是通过栈来实现的。算术表达式(a−b)*c+d 首先计算 a−b，因此 a、b 先入

栈，当遇到“−”时出栈，得出结果并压入栈；然后计算与 c 相乘的结果，将 c 入栈，遇到“*”时，出栈，计算乘积并将结果入栈；最后进行与 d 的加法运算，将 d 入栈，遇到“+”时出栈，计算结果。

【答案】C

【典型题 2-19】算术表达式 x−(y+c)*8 的后缀式是__(22)__(−、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例)。(2012 年 5 月真题 22)

(22) A. xyc8 - + *　　B. xy−c+8*　　C. xyc8*+−　　D. xyc+8* −

【解析】后缀表达式不包含括号，运算符放在两个运算对象的后面，所有的计算按运算符出现的顺序，严格从左向右进行(不再考虑运算符的优先规则)，如(2 + 1) * 3 的后缀式为 21 + 3*。运用后缀表达式进行计算的具体做法：建立一个栈 S，从左到右读后缀表达式，如果读到操作数就将它压入栈 S 中，如果读到 n 元运算符(即需要参数个数为 n 的运算符)则取出由栈顶向下的 n 项按操作符运算，再将运算的结果代替原栈顶的 n 项，压入栈 S 中。如果后缀表达式未读完，则重复上面过程，最后输出栈顶的数值则为结束。

【答案】D

2.5　过 关 习 题

1. 以下关于可视化程序设计的叙述中，错误的是______。

　A. 可视化程序设计使开发应用程序无须编写程序代码

　B. 可视化程序设计基于面向对象的思想，引入了控件和事件驱动

　C. 在可视化程序设计中，构造应用程序界面就像搭积木

　D. 在可视化程序设计中，采用解释方式可随时查看程序的运行效果

2. 以下关于变量和常量的叙述中，错误的是______。

　A. 变量的取值在程序运行过程中可以改变，常量则不行

　B. 变量具有类型属性，常量则没有

　C. 变量具有对应的存储单元，常量则没有

　D. 可以对变量赋值，不能对常量赋值

3. 若 C 程序的表达式中引用了未赋初值的变量，则______。

　A. 编译时一定会报告错误信息，该程序不能允许

　B. 可以通过编译并运行，但运行时一定会报告异常

　C. 可以通过编译，但链接时一定会报告错误而不能运行

　D. 可以通过编译并运行，但运行结果不一定是期望的结果

4. 若一种程序设计语言规定其程序中的数据必须具有类型，则有利于______。

① 在翻译程序的过程中为数据合理分配存储单元

② 对参与表达式计算的数据对象进行检查

③ 定义和应用动态数据结构

④ 规定数据对象的取值范围及能够进行的运算

⑤ 对数据进行强制类型转换

　A. ①②③　　B. ①②④　　C. ②④⑤　　D. ③④⑤

5. 函数 t、f 的定义如下所示，其中，a 是整型全局变量。设调用函数 t 前 a 的值为 5，则在函数中以

传值调用(call by value)方式调用函数 f 时，输出为 (1) ；在函数 t 中以引用调用(call by reference)方式调用函数时，输出为 (2) 。

```
t( ):  int x=f(a);        f(int r):  a=r+1;  r=r*2;
       print a+x;                    return r;
```

(1) A. 12　B. 16　C. 20　D. 24

(2) A. 12　B. 16　C. 20　D. 24

6. 传值与传地址是函数调用时常采用的信息传递方式，______。

A. 在传值方式下，是将形参的值传给实参

B. 在传值方式下，形参可以是任意形式的表达式

C. 在传地址方式下，是将实参的地址传给形参

D. 在传地址方式下，实参可以是任意形式的表达式

7. 以下关于汇编语言的叙述中，错误的是______。

A. 汇编语言源程序中的指令语句将被翻译成机器代码

B. 汇编程序先将源程序中的伪指令翻译成机器代码，然后再翻译成指令语句

C. 汇编程序以汇编语言源程序为输入，以机器语言表示的目标程序为输出

D. 汇编语言的指令语句必须具有操作码字段，可以没有操作数字段

8. 编译程序分析源程序的阶段依次是______。

A. 词法分析、语法分析、语义分析　B. 语法分析、词法分析、语义分析

C. 语义分析、语法分析、词法分析　D. 语义分析、词法分析、语法分析

9. 算术表达式采用逆波兰式表示时不用括号，可以利用 (1) 进行求值。与逆波兰式 ab−cd+* 对应的中缀表达式是 (2) 。

(1) A. 数组　B. 栈　C. 队列　D. 散列表

(2) A. a−b+c*d　B. (a−b)*c+d　C. (a−b)*(c+d)　D. a−b*c+d

10. 以下关于高级程序设计语言翻译的叙述中，正确的是______。

A. 可以先进行语法分析，再进行词法分析

B. 在语法分析阶段可以发现程序中的所有错误

C. 语义分析阶段的工作与目标机器的体系结构密切相关

D. 目标代码生成阶段的工作与目标机器的体系结构密切相关

11. 图 2-7 所示为一个有限自动机(其中，A 是初态、C 是终态)，该自动机可识别______。(2011 年 5 月上午试题 49)

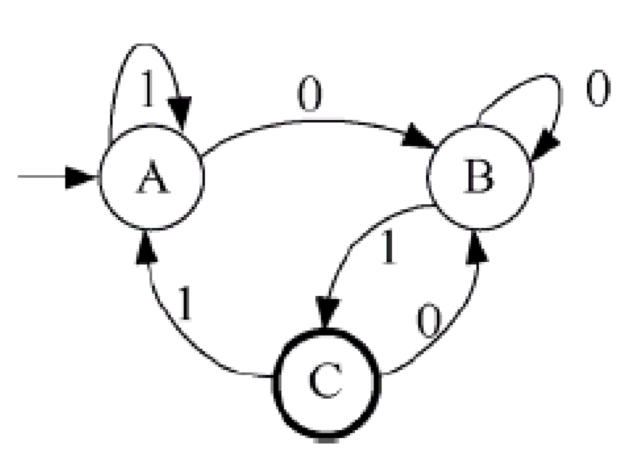


图 2-7　有限自动机的状态转换图

A. 0000　B. 1111　C. 0101　D. 1010

12. 图 2-8 所示为两个有限自动机 M1 和 M2 (A 是初态、C 是终态)，______。(2010 年 11 月上午试题 48)

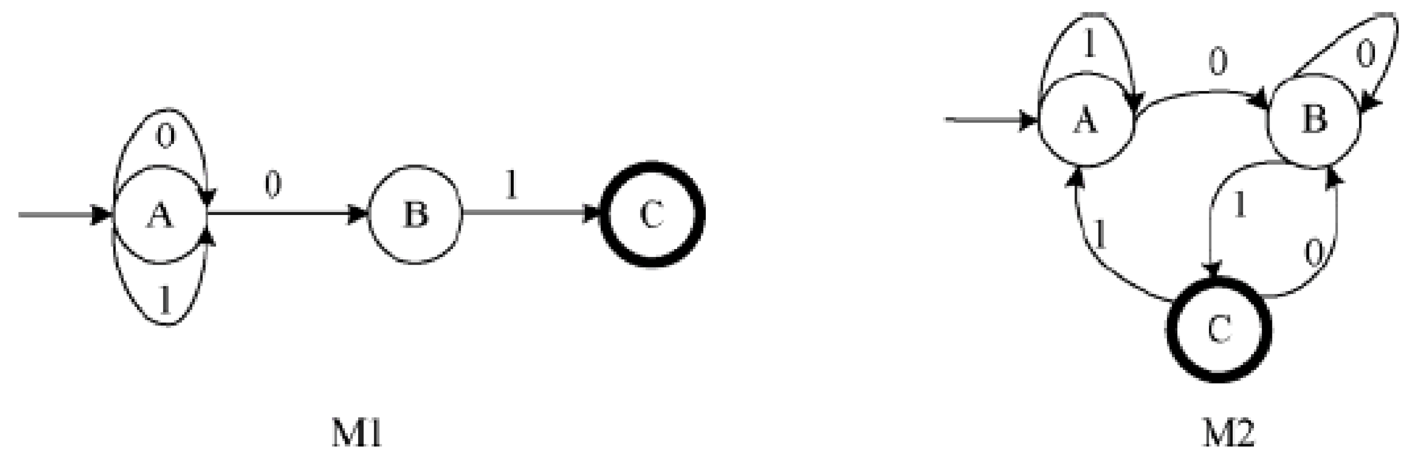


图 2-8　两个有限自动机的状态转换图

A. M1 和 M2 都是确定的有限自动机

B. M1 和 M2 都是不确定的有限自动机

C. M1 是确定的有限自动机，M2 是不确定的有限自动机

D. M1 是不确定的有限自动机，M2 是确定的有限自动机

13. 图 2-9 所示为一个有限自动机(其中，A 是初态，C 是终态)，该自动机识别的语言可用正规式______表示。(2011 年 11 月上午试题 48)

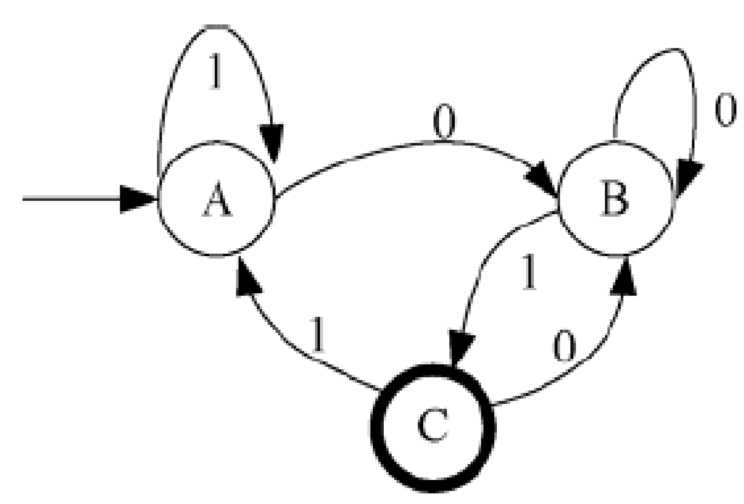


图 2-9　有限自动机的状态转换图

A. (0|1)*01　　B. 1*0*10*1　　C. 1*(0)*01　　D. 1*(0|10)*1*

第 3 章

操 作 系 统

本章要点

- 操作系统基础知识
- 处理机管理
- 存储管理
- 设备管理
- 文件管理

3.1 操作系统基础知识

考核说明：本节主要介绍操作系统的基本概念，包括操作系统的定义、作用、类型，了解即可。

1. 操作系统的定义

操作系统(Operating System，OS)是计算机系统中的一个系统软件，它可以管理和控制计算机系统的硬件和软件资源，合理地组织计算机的工作流程，控制程序的执行，并且向用户提供一个良好的工作环境和友好的接口。

2. 操作系统的作用

操作系统具有如下作用。

(1) 通过资源管理，提高计算机系统的效率。

(2) 改善人机界面，向用户提供友好的工作环境。

3. 操作系统的特征

操作系统主要有并发性、共享性、虚拟性和不确定性四个基本特征。

4. 操作系统的功能

操作系统具有以下功能。

- 处理机管理。处理机管理实际上是指对处理机执行“时间”的管理，采用多道程序等技术将 CPU 真正合理地分配给每个任务。常用的资源管理单位有进程和线程。
- 文件管理。文件管理(信息管理)包括文件存储空间管理、目录管理、文件的读写管理和存取控制、软件管理等。
- 存储管理。存储管理主要是对主存空间进行管理。
- 设备管理。设备管理的目标是方便设备使用，提高 CPU 与 I/O 设备的利用率。
- 作业管理。作业管理包括任务、界面管理、人机交互、图形界面、语音控制和虚拟现实等。

5. 操作系统的类型

操作系统可分为以下类型。

- 批处理操作系统。
- 分时操作系统。
- 实时操作系统。
- 网络操作系统。
- 分布式操作系统。
- 微机操作系统。
- 嵌入式操作系统。

真题链接

【例 3-1】设计操作系统时不需要考虑的问题是 (23) 。(2014 年 5 月真题 23)

(23) A. 计算机系统中硬件资源的管理　　B. 计算机系统中软件资源的管理

C. 用户与计算机之间的接口　　D. 语言编译器的设计实现

【解析】明白操作系统的作用，就不难选出结果。操作系统有两个重要作用：①通过资源管理提高计算机系统的效率，操作系统是计算机系统的资源管理者，它含有对系统软/硬件资源实施管理的一组程序。②改善人机界面，向用户提供友好的工作环境。

【答案】D

3.2 处理机管理

考核说明：经常会考到进程的基本概念、进程间的通信、管程、进程调度、死锁和线程，其中，PV 操作和死锁的处理是重点也是难点，要能透彻理解。

3.2.1 基本概念

1. 前驱图

前驱图是一个有向无循环图，图由结点和结点间的有向边组成，结点代表各程序段的操作，而结点间的有向边表示两程序段操作之间存在的前驱关系(“→”)。两程序段 P_i 和 P_j 的前驱关系表示成 $P_i \rightarrow P_j$，其中 P_i 是 P_j 的前驱，P_j 是 P_i 的后继，其含义是 P_i 执行完毕才能执行 P_j。

2. 进程

进程通常是由程序、数据及进程控制块(PCB)组成的。进程的程序部分描述了进程需要完成的功能，进程数据集合部分包括程序执行时所需的数据及工作区。

进程控制块是进程的描述信息和控制信息，是进程动态特性的集中反映，也是进程存在的唯一标志。进程控制块包含的主要内容有进程标志符、状态、位置信息、控制信息、队列指针、优先级、现场保护区及其他。PCB 是操作系统中最主要的数据结构之一，既是进程存在的标志和调度的依据，又是进程可以被打断并能恢复运行的基础。操作系统通过 PCB 管理进程，一般 PCB 是常驻主存的，尤其是调度信息必须常驻主存。

3. 进程的状态及其转换

三态模型中最基本的状态有三种：运行、就绪和阻塞。

在进程运行过程中，由于自身进展情况及外界环境的变化，这三种基本状态可以在一定的条件下相互转换。进程的状态及转换如图 3-1 所示。

五态模型在三态模型的基础上增加了新建态和终止态。

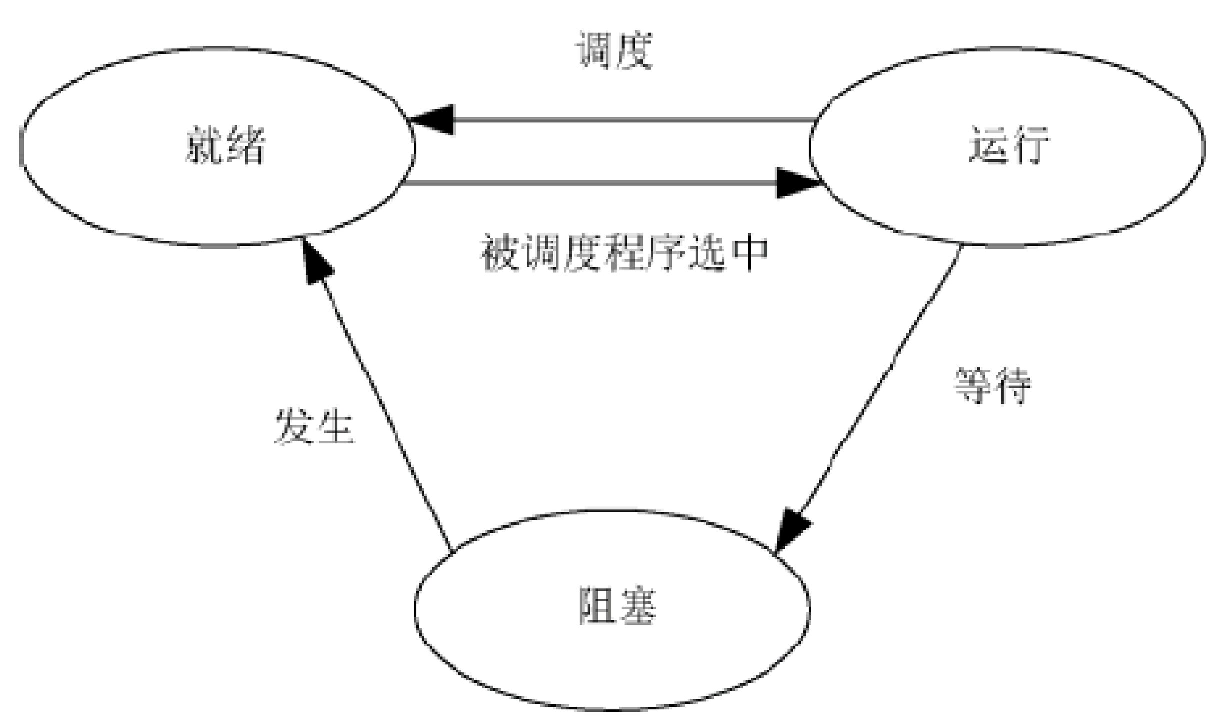


图 3-1 进程的状态及其转换

真题链接

【例 3-2】在如图 3-2 所示的进程资源图中，(27)；该进程资源图是(28)。(2014 年 11 月真题 27、28)

(27) A. P1、P2、P3 都是阻塞节点

B. P1 是阻塞节点，P2、P3 是非阻塞节点

C. P1、P2 是阻塞节点，P3 是非阻塞节点

D. P1、P2 是非阻塞节点，P3 是阻塞节点

(28) A. 可以化简的，其化简顺序为 P1→P2→P3

B. 可以化简的，其化简顺序为 P3→P1→P2

C. 可以化简的，其化简顺序为 P2→P1→P3

D. 不可以化简的，因为 P1、P2、P3 申请的资源都不能得到满足

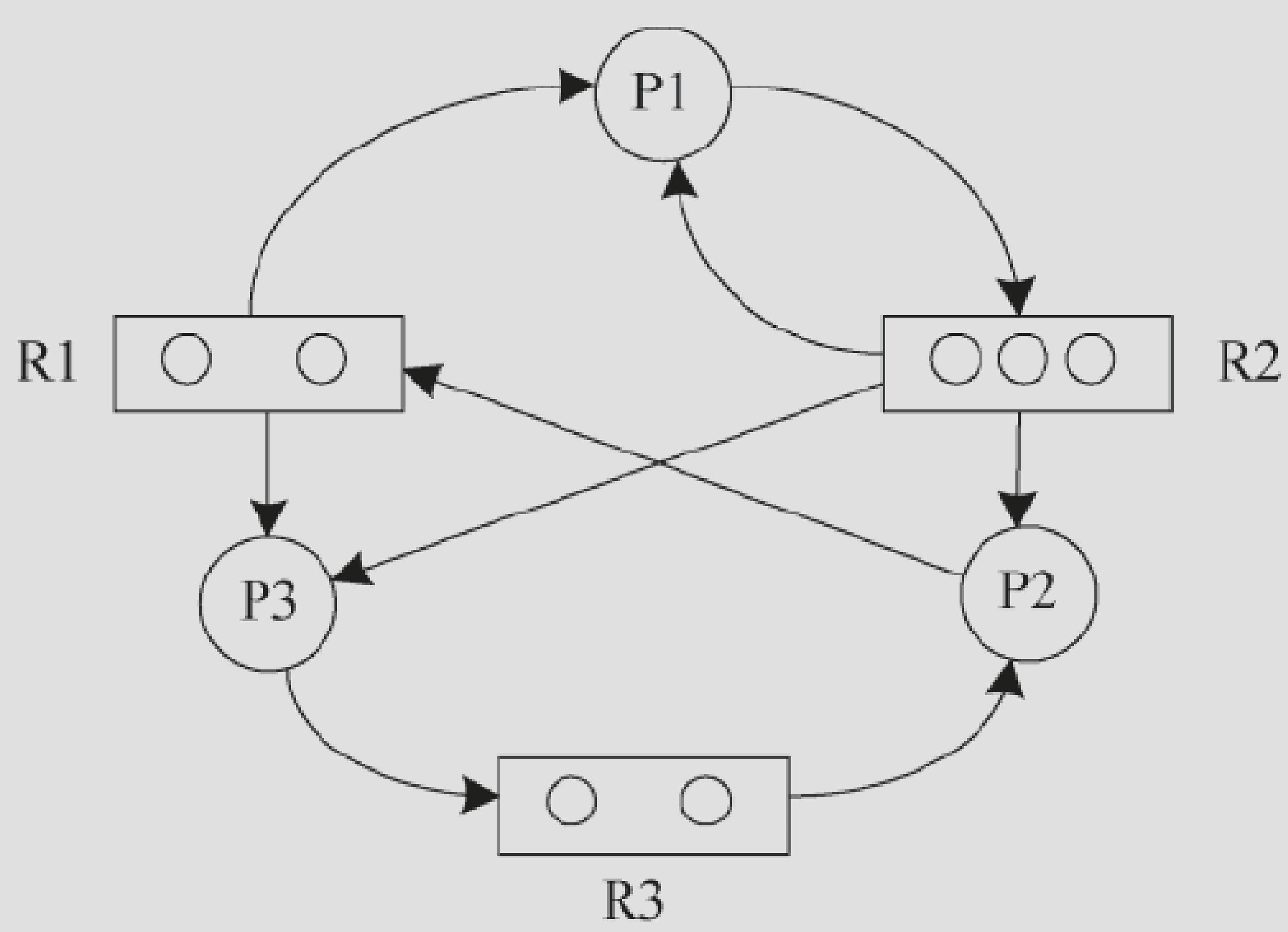


图 3-2 进程资源图

【解析】R1 已经全部分配给 P1 和 P3，所以 P2 再请求一个 R1 的时候，将进入阻塞状态。同理，R2 已经全部分配给 P1、P2 和 P3，当 P1 再请求一个 R2 时，将陷入阻塞。R3 还有一个未用资源，当 P3 申请时，可以顺利获得，故不会阻塞。

因为 P3 非阻塞且非孤立，所以可以化简。将其所用资源归还后，P1 获得 R2，即可运行，然后也可以化简，最后 P2 可以运行。

【答案】(27)C；(28)B

3.2.2 进程间的通信

1. 同步与互斥

同步是合作进程间的直接制约问题，互斥是申请临界资源进程间的间接制约问题。

1) 同步

相互合作的进程需要在某些确定点上协调它们的工作，当一个进程到达这些点后，除非另一个进程已经完成某些操作，否则就不得不停下来等待这些操作结束。这就是进程间的同步。

2) 互斥

在多道程序系统中，各进程可以共享各类资源，但有些资源一次只能供一个进程使用，称为临界资源。这就产生了进程间的间接制约问题：互斥。

3) 临界区管理的原则

临界区是进程中对临界资源实施操作的那段程序。互斥临界区管理的原则是：有空即进，无空则等；有限等待，让权等待。

2. 信号量与 PV 操作

信号量机制主要有整型信号量、记录性信号量、信号量集机制。

1) 整型信号量与 PV 操作

信号量是一个整型变量，根据控制对象的不同赋不同的值。信号量可分为两类。

- 公用信号量：实现进程间的互斥，初值=1 或资源的数目。
- 私用信号量：实现进程间的同步，初值=0 或某个正整数。

信号量 S 的物理意义为：$S\geqslant 0$，表示某资源的可用数；$S<0$，其绝对值表示阻塞队列中等待该资源的进程数。PV 操作是实现进程同步与互斥的常用方法。

P 操作定义：$S:=S-1$，若 $S\geqslant 0$，则执行 P 操作的进程继续执行；否则，若 $S<0$，则置该进程为阻塞状态，并将其插入阻塞队列。

V 操作定义：$S:=S+1$，若 $S>0$，则执行 V 操作的进程继续执行；否则，若 $S\leqslant 0$，则从阻塞状态唤醒一个进程，并将其插入就绪队列，执行 V 操作的进程继续执行。

2) 利用 PV 操作实现进程的互斥

令信号量的初值为 1，当进程进入临界区时执行 P 操作，退出临界区时执行 V 操作。

3) 利用 PV 操作实现进程的同步

进程的同步是由于进程间合作而引起的相互制约的问题。要实现进程的同步，可用一个信号量与消息联系起来。当信号量的值为 0 时表示消息未产生，当信号量的值为非 0 时表示希望的消息已经存在。假定用信号量 S 表示某条消息，进程可以通过调用 P 操作测试消息是否达到，调用 V 操作通知消息已经准备好。

3. 高级通信原语

PV 操作是用来协调进程间关系的，编程较困难、效率低，通信对用户不透明，生产者每次只能向缓冲区放一个消息，消费者只能从缓冲区中取一个消息。所以交换的信息量多

时要引入高级通信原语。进程高级通信的类型主要有如下几种。

(1) 共享存储系统：相互通信的进程共享某些数据结构或存储区，以实现进程之间的通信。

(2) 消息传递系统：进程间的数据交换以消息为单位，程序员直接利用系统提供的一组通信命令(原语)来实现通信，如 Send(A)、Receive(A)。

(3) 管道通信：所谓管道，是指用于连接两个进程之间的一个打开的共享文件(pipe 文件)。向管道(共享文件)提供输入的发送进程(即写进程)，以字符流的形式将大量的数据送入管道；而接收进程可从管道的另一端接收大量的数据。由于通信时采用管道，因此叫管道通信。

真题链接

【例 3-3】假设系统采用 PV 操作实现进程同步与互斥，若 n 个进程共享两台打印机，那么信号量 S 的取值范围为 (23) 。(2014 年 11 月真题 23)

(23) A. $-2\sim n$　B. $-(n-1)\sim 1$　C. $-(n-1)\sim 2$　D. $-(n-2)\sim 2$

【解析】信号量初值等于资源数量，即为 2，由于同时最多有 2 个进程访问打印机，其余进程必须处理等待状态，故 S 的最小值为 $-(n-2)$。

【答案】D

3.2.3 进程调度

1. 调度方式

调度方式是指当有更高优先级的进程到来时如何分配 CPU。调度方式分为可剥夺式和不可剥夺式两种。可剥夺式是指当有更高优先级的进程到来时，强行将正在运行的进程所占用的 CPU 分配给高优先级的进程；不可剥夺式是指当有更高优先级的进程到来时，必须等待正在运行的进程自动释放占用的 CPU，然后才能将 CPU 分配给高优先级的进程。

2. 进程调度算法

常用的进程调度算法有先来先服务、时间片轮转、优先级调度和多级反馈调度算法。

3.2.4 死锁

所谓死锁是指两个以上的进程互相都因要求对方已经占有的资源，而导致无法运行下去的现象。

1. 产生死锁的原因

产生死锁的原因是资源竞争及进程推进顺序非法。

2. 产生死锁的必要条件

产生死锁的必要条件如下。

- 互斥条件：进程对其要求的资源进行排他性控制，即一次只允许一个进程使用。
- 请求保持条件：零星地请求资源，即已获得部分资源后又请求资源被堵塞。
- 不可剥夺条件：进程已获得资源在未使用完之前不能被剥夺，只能在使用完时由自己释放。
- 环路条件：发生死锁时，在进程资源有向图中必构成环路，其中每个进程占有下一个进程申请的一个或多个资源。

3. 死锁的处理

下面介绍死锁的处理。

- 死锁的预防。根据产生死锁的四个必要条件，只要使其中之一不能成立，死锁就不会出现。为此，可以采取的预防措施有：预先静态分配法和资源有序分配法。
- 死锁的避免。最著名的死锁避免算法是 Dijkstra 提出的银行家算法，其思想是：对于进程发出的每一个系统可以满足的资源请求命令加以检测，如果发现分配资源后，系统可能进入不安全状态，则不予分配；若分配资源后系统仍处于安全状态，则分配资源。与死锁预防策略相比提高了资源的利用率，但增加了系统的开销。
- 死锁的检测。这种方法对资源的分配不加限制，即允许死锁发生。但系统定时地运行一个“死锁检测”程序，判断系统是否发生死锁，若检测到有死锁，则设法加以解除。
- 死锁的解除。检测到死锁发生后，常采用资源剥夺法和撤销进程法解除死锁。

真题链接

【例 3-4】假设某计算机系统中资源 R 的可用数为 6，系统中有 3 个进程竞争 R，且每个进程都需要 *i* 个 R，则该系统可能会发生死锁的最小 *i* 值是__(24)__。若信号量 S 的当前值为−2，则 R 的可用数和等待 R 的进程数分别为__(25)__。(2014 年 5 月真题 24～25)

(24) A. 1　　B. 2　　C. 3　　D. 4

(25) A. 0、0　　B. 0、1　　C. 1、0　　D. 0、2

【解析】如果 i=1，即每个进程都需要 1 个 R，3 个进程同时运行需要 3 个 R，还剩 3 个 R，不会发生死锁。如果 i=2，即每个进程都需要 2 个 R，3 个进程同时运行需要 6 个 R，而 R 的可用数正好为 6，不会发生死锁。如果 i=3，即每个进程都需要 2 个 R，当 3 个进程分别占有 2 个 R 时，都需要再申请一个 R 资源才能正常运行，但此时已经没有 R 资源了，进程之间便出现了相互等待的状况，发生死锁。

信号量的值小于 0，表示没有可用的资源，其绝对值表示阻塞队列中等待该资源的进程数。

【答案】(24)C；(25)D

3.2.5　线程

线程是比进程更小的能独立运行的基本单位。在引入线程的操作系统中，线程是进程中的一个实体，是系统独立分配和调度的基本单位。线程只拥有一点在运行中必不可少的资源(如程序计数器、一组寄存器和栈)，但它可与同属一个进程的其他线程共享该进程所占用的全部资源。一个线程可以创建或撤销另一个线程，同一个进程中的多个线程之间可以

并发执行。线程也同样有就绪、等待和运行三种基本状态。

3.3 存 储 管 理

考核说明：本节主要介绍与存储管理相关的概念、各种存储管理方案，后者较为重要，要能理解各种存储管理方案的实现方法。

3.3.1 基本概念

1. 存储器的结构

存储器的功能是保存数据，存储器的发展方向是高速度、大容量和小体积。一般存储器的结构有“寄存器—主存—外存”结构或“寄存器—缓存—主存—外存”结构。

2. 地址重定位

地址重定位是指程序的逻辑地址被转换成主存的物理地址的过程。在可执行文件装入时需要解决可执行文件中地址(指令和数据)和主存地址的对应关系，由操作系统中的装入程序 Loader 和地址重定位机构来完成。地址重定位分为静态地址重定位和动态地址重定位。

- 静态地址重定位：指当用户程序被装入主存时已经实现了逻辑地址到物理地址的变换，在程序执行期间不再发生变化。
- 动态地址重定位：指在程序运行期间完成逻辑地址到物理地址的变换。其实现依赖于硬件地址变化机构，如基地址寄存器(BR)。

3. 存储管理方案

存储管理的主要目的是解决多个用户使用主存的问题，其存储管理方案主要包括分区存储管理、分页存储管理、分段存储管理、段页式存储管理以及虚拟存储管理。

3.3.2 分区存储管理

1. 固定分区

固定分区是一种静态分区方式，在系统生成时已将主存分成若干个分区，每个分区的大小可以不等，但分区大小固定不变，每个分区装一个且只能装一个作业。操作系统通过主存分配情况表管理主存。

2. 可变分区

可变分区是一种动态分区方式，存储空间的划分是在作业装入时进行的，故分区的个数是可变的，分区的大小刚好等于作业的大小。

引入可变分区方法，可以使主存分配有较大的灵活性，也提高了主存的利用率；但是可变分区会引起碎片的产生。

3. 可重定位分区

可重定位分区是解决碎片问题的简单而又行之有效的方法。其基本思想是移动所有已分配好的分区，使之成为连续区域。由于移动分区是要付出代价的，因此通常是在用户请求空间得不到满足时进行。移动已分配的分区会导致地址发生变化，所以会产生地址重定位的问题。

真题链接

【例 3-5】假设内存管理采用可变式分区分配方式，系统中有五个进程 P1～P5，且某一时刻内存的使用情况如图 3-3 所示(图中空白处表示未使用分区)。此时，若 P5 进程运行完并释放其占有的空间，则释放后系统的空闲区数应 (27) ；造成这种情况的原因是 (28) 。(2013 年 5 月真题 27～28)

分区号	进程
0	P1
1	P2
2	
3	P4
4	P3
5	
6	P5
7	

图 3-3　内存使用情况

(27)　A. 保持不变　　B. 减 1　　C. 加 1　　D. 置零

(28)　A. 无上邻空闲区，也无下邻空闲区　　B. 有上邻空闲区，但无下邻空闲区

　　C. 有下邻空闲区，但无上邻空闲区　　D. 有上邻空闲区，也有下邻空闲区

【解析】可变分区是一种动态分区方式，在作业执行前并不建立分区，而是在处理作业过程中按需要建立分区。当P5运行完后释放空间时，发现其释放的空间上下方都有空闲区，故将两个空闲区与自身要释放的空闲区合并，从而形成一个空闲区，导致系统的空闲区数量上减1。而造成这种现象的直接原因就是要释放的空闲区相邻的上下方空闲区。

【答案】(27)B；(28)D

3.3.3　分页存储管理

1. 纯分页存储管理

1)　分页原理

系统将进程的地址空间划分成若干个大小相等的区域，称为页。同样地，将主存空间划分成与页相同大小的若干物理块，称为块或页框。在为进程分配主存时，将进程中的若干页分别装入多个不相邻接的块中。

2)　地址结构

分页系统的地址结构如图 3-4 所示，它由两部分组成：前一部分为页号 P；后一部分为偏移量 W，即页内地址。图中的地址长度为 32 位，其中 0～11 位为页内地址(每页的大小为

4KB)，12～31 位为页号，所以允许地址空间的大小最多为 1MB 个页。

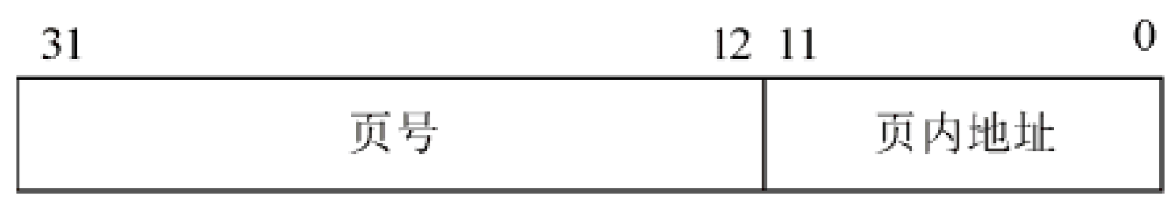


图 3-4　分页系统的地址结构

3)　页表

每个页在页表中占一个表项，记录该页在主存中对应的物理块号。进程在执行时，通过查找页表就可以找到每页所对应的物理块号。可见，页表的作用是实现从页号到物理块号的地址映射。

2. 快表

在地址映射过程中，共需访问主存两次。第一次访问页表，得到数据的物理地址；第二次才是存取数据。为了提高访问主存的速度，可以采取两种方法：一种方法是在地址映射机制中增加一组高速寄存器保存页表，这需要大量的硬件开销，经济上不可行；另一种方法是在地址映射机制中增加一个小容量的联想寄存器(相联存储器)，它由一组高速寄存器组成，称为快表。快表用来存放当前访问最频繁的少数活动页的页号及相关信息。

真题链接

【例 3-6】某计算机系统页面大小为 4KB，若进程的页面变换表如图 3-5 所示，逻辑地址为十六进制 1D16H。该地址经过变换后，其物理地址应为十六进制_(26)_。(2014 年 5 月真题 26)

页号	物理块号
0	1
1	3
2	4
3	6

图 3-5　页面变换表

(26) A. 1024H　　B. 3D16H　　C. 4D16H　　D. 6D16H

解　析：页面大小为 4KB，而 $4K=2^{12}$，因此逻辑地址的低 12 位对应页内地址，高位对应页号。一位十六进制数对应 4 位二进制数，3 位十六进制数则对应 12 位二进制数，题目中逻辑地址为十六进制 1D16H，因此 D16H 为页内地址，页号为 1。查页面变换表，页号 1 对应的物理块号为 3，将物理块号与页内地址 D16H 拼接起来即可得到物理地址 3D16H。

【答案】B

3.3.4　分段存储管理

1. 基本原理

在分段存储管理方式中，作业的地址空间按程序自身的逻辑关系划分为若干个程序段，每个段是一组完整的逻辑信息。每个段都有自己的段名，且有一个段号。段号从 0 开始，

每一段也从 0 开始编址，段内地址是连续的，各段长度是不等的。

分段系统的逻辑地址由段号(名)和段内地址两部分组成。在该地址结构中，允许一个作业最多有 2^{16} 个段，每个段的最大长度为 64KB。其地址结构如图 3-6 所示。

31　　　　16	15　　　　0
段号 s	段内地址 d

图 3-6　分段的地址结构

在分段式存储管理系统中，为每个段分配一个连续的分区，而进程中的各个段可以离散地分配到主存中不同的分区中。在系统中为每个进程建立一张段映射表，简称为“段表”。段表实现了从逻辑段到物理主存区的映射。

2. 分段系统的地址变换

为了实现从逻辑地址到物理地址的变换功能，系统中设置了段表寄存器，用于存放段表基址和段表长度。在进行地址变换时，系统对逻辑地址中的段号与段表长度进行比较。

段是信息的逻辑单位，因此分段系统的一个突出优点是易于实现段的共享，即允许若干个进程共享一个或多个段，而且对段的保护也十分简单。在分页系统中，虽然也能实现程序和数据的共享，但远不如分段系统来得方便。

3.3.5　段页式存储管理

段页式存储管理结合了段式存储管理和页式存储管理的优点，克服了两者的缺点。其基本原理是：先将整个主存划分成大小相同的存储块，将用户程序按程序的逻辑关系分为若干个段，并为每个段赋予一个段名，再将每个段划分成若干个页，以页架为单位离散分配。

在段页式系统中，其地址结构由段号、段内页号及页内地址三部分组成，如图 3-7 所示。因此，系统中需同时配置段表和页表。由于允许将一个段中的页进行离散分配，因此使段表的内容略有变化：它不再是段的主存基址和段长，而是页表始址和页表长度。

段号 s	段内页号 p	页内地址 w

图 3-7　段页式管理的地址结构

在段页式系统中，为了便于实现地址变换，需要配置一个段表寄存器，存放段表基址和段表长度。

在进行地址变换时，首先利用段号 s，将它与段表长度 TL 进行比较。若 s<TL，表示未越界，于是利用段表基址和段号来求出该段对应的段表项在段表中的位置，从中得到该段的页表基址，并利用逻辑地址中的段内页号 p 来获得对应页的页表项位置，从中读出该页所在的物理块号 b，再利用块号 b 和页内地址构成物理地址。

真题链接

【例 3-7】假设段页式存储管理系统中的地址结构如图 3-8 所示，则系统<u>(24)</u>。(2014 年 11 月真题 24)

31　　　　22	21　　　　12	11　　　　0
段 号	页 号	页内地址

图 3-8　段页式存储管理系统中的地址结构

(24) A. 最多可有 2048 个段，每个段的大小均为 2048 个页，页的大小为 2K
　　B. 最多可有 2048 个段，每个段最大允许有 2048 个页，页的大小为 2K
　　C. 最多可有 1024 个段，每个段的大小均为 1024 个页，页的大小为 4K
　　D. 最多可有 1024 个段，每个段最大允许有 1024 个页，页的大小为 4K

【解析】段号数量 2^10 = 1024，段内最大页数 2^10=1024，页大小 2^12=4096 字节。

【答案】D

3.3.6　虚拟存储管理

1. 虚拟存储器的引入

1)　局部性原理

局部性表现为时间局部性和空间局部性两类。

2)　虚拟存储器的定义

根据局部性原理，一个作业在运行之前，没有必要把作业全部装入主存，而仅将当前要运行的那部分页面或段先装入主存启动运行，其余部分暂时留在磁盘上。

程序在运行时如果它所要访问的页(段)已调入主存，便可继续执行下去；但如果程序所要访问的页(段)尚未调入主存(称为缺页或缺段)，程序应利用操作系统所提供的请求调页(段)功能，将它们调入主存，以使进程能继续执行下去。

如果此时主存已满，无法再装入新的页(段)，则还要再利用页(段)的置换功能，将主存中暂时不用的页(段)调出至磁盘上，腾出足够的主存空间后，再将所要访问的页(段)调入主存，使程序继续执行下去。这样，便可使一个大的用户程序在较小的主存空间中运行，也可使主存中同时装入更多的进程并发执行。从用户角度看，该系统所具有的主存容量，将比实际主存容量大得多，人们把这样的存储器称为虚拟存储器。

3)　虚拟存储器的功能

虚拟存储器具有请求调入功能和置换功能，能仅把作业的一部分装入主存便可运行作业，能从逻辑上对主存容量进行扩充。其逻辑容量由主存和外存容量之和以及 CPU 可寻址的范围来决定，其运行速度接近于主存速度。所以说，虚拟存储技术是一种性能非常优越的存储器管理技术，被广泛地应用于大、中、小型和微型机中。

4)　虚拟存储器的实现

(1)　请求分页系统：在分页系统的基础上，增加了请求调页功能和页面置换功能所形成的页式虚拟存储系统。

(2)　请求分段系统：在分段系统的基础上，增加了请求调段和分段置换功能所形成的段式虚拟存储系统。

(3)　请求段页式系统：在段页式系统的基础上，增加了请求调页和页面置换功能形成的段页式虚拟存储系统。

5)　虚拟存储器的特征

虚拟存储器的特征包括离散性、多次性、对换性和虚拟性。

2. 请求分页管理的实现

请求分页系统是在纯分页系统的基础上，增加了请求调页功能、页面置换功能所形成的页式虚拟存储系统，是目前常用的一种虚拟存储器方式。

请求分页的页表机制是在纯分页的页表机制上形成的，由于只将应用程序的一部分调入主存，还有一部分仍在磁盘上，因此需要在页表中再增加若干项，如状态位、访问字段和辅存地址等供程序(数据)在换进、换出时参考。

请求分页系统中的地址变换机构，是在分页系统的地址变换机构的基础上增加了某些功能，如产生和处理缺页中断，从主存中换出一页实现虚拟存储。

在请求分页系统中，每当所要访问的页面不在主存时，便要产生一个缺页中断，请求 OS 将所缺的页调入主存，这是由缺页中断机构完成的。

3. 页面置换算法

请求分页系统的核心问题是选择合适的页面置换算法。常用的页面置换算法如下。

(1)　最佳(Optimal)置换算法。它是一种理想化的算法，性能最好，但在实际上难以实现，通常用来评价其他算法。

(2)　先进先出(FIFO)置换算法。该算法总是淘汰最先进入主存的页面，即选择在主存中驻留时间最久的页面予以淘汰。这是一种最直观，也是性能最差的算法，会出现贝来迪(Belady)异常现象，即如果对一个进程未分配它所要求的全部页面，有时就会出现分配的页面数增多但缺页率反而提高的异常现象。

(3)　最近最久未使用置换算法。该算法是选择最近最久未使用的页面予以淘汰，在实现时需要硬件的支持(寄存器或栈)。

(4)　最近未用置换算法。将最近一段时间未引用过的页面换出，是一种 LRU 的近似算法。

3.4　设 备 管 理

考核说明：本节主要介绍 I/O 软件、设备管理采用的缓冲技术、磁盘调度，其中磁盘调度是重要考点，要理解和掌握各种磁盘调度算法。

3.4.1　设备管理概述

1. 设备的分类

1)　按数据组织分类

按数据组织分类，设备可分为块设备和字符设备。

2)　按资源分配的角度分类

按资源分配的角度分类，设备可分为独占设备、共享设备和虚拟设备。

3) 按数据传输率分类

按数据传输率分类，设备可分为低速设备、中速设备和高速设备。

4) 其他分类方法

按输入输出对象分类，设备可分为人机通信设备和机机通信设备。

按是否可交互分类，设备可分为非交互设备和交互设备。

2. 设备管理的目标与任务

1) 设备管理的目标

设备管理的目标主要是如何提高设备的利用率，为用户提供方便统一的界面。

2) 设备管理的任务

设备管理的任务是保证在多道程序环境下，当多个进程竞争使用设备时，按一定策略分配和管理各种设备，控制设备的各种操作，完成输入/输出设备与主存之间的数据交换。

3.4.2 I/O 软件

I/O 设备管理软件一般分为 4 层：中断处理程序、设备驱动程序、与设备无关的系统软件和用户层 I/O 软件。至于一些具体分层时细节上的处理，是依赖于系统的，没有严格的划分，只要有利于设备独立这一目标，就可以为了提高效率而设计不同的层次结构。

中断处理程序的作用是，当 I/O 操作结束时，唤醒驱动程序。中断处理过程如下。

(1) CPU 检查响应中断的条件是否满足。

(2) 如果 CPU 响应中断，则 CPU 关中断，使其进入不可再次响应中断的状态。

(3) 保存被中断进程的现场。

(4) 分析中断原因，调用中断处理子程序。

(5) 执行中断处理子程序。

(6) 退出中断，恢复被中断进程的现场或调度新进程占据 CPU。

(7) 开中断，CPU 继续执行。

3.4.3 设备管理采用的相关缓冲技术

1. 通道技术

引入通道的目的是使数据的传输独立于 CPU，使 CPU 从繁重的 I/O 工作中解脱出来。设置通道后，CPU 只需向通道发出 I/O 命令，通道收到命令后，从主存中取出本次 I/O 要执行的通道程序，并执行，仅当通道完成 I/O 任务后，才向 CPU 发出中断信号。

根据信息交换方式的不同，将通道分为字节多路通道、数组选择通道和数组多路通道三种。

2. 直接存储访问方式

直接存储访问(Direct Memory Access，DMA)是指数据在主存和 I/O 设备间传送一个数据块的过程中，不需要 CPU 的任何干涉，只需要 CPU 在过程开始启动与过程结束时的处理，

实际操作由 DMA 硬件直接执行完成，CPU 在此传送过程中可做其他事情。

3. 缓冲技术

缓冲技术可提高外设的利用率，尽可能使外设处于忙状态。缓冲技术可以分为硬件缓冲和软件缓冲技术。硬件缓冲是利用专门的硬件寄存器作为缓冲，软件缓冲是通过操作系统来管理的。引入缓冲技术的主要原因有以下几个方面。

(1) 缓和 CPU 与 I/O 设备间速度不匹配的矛盾。

(2) 减少对 CPU 的中断频率，放宽对中断响应时间的限制。

(3) 提高 CPU 和 I/O 设备之间的并行性。

因为在所有的 I/O 设备与处理机之间，都使用了缓冲区来交换数据，所以操作系统必须组织和管理好这些缓冲区。缓冲可以分为单缓冲、双缓冲、多缓冲和环形缓冲。

4. SPOOLing 技术

所谓 SPOOLing 技术实际上是用一类物理设备模拟另一类物理设备的技术，是使独占使用的设备变成多台虚拟设备的一种技术，也是一种速度匹配技术。SPOOLing 系统是由“预输入程序”、“缓输出程序”、“井管理程序”以及输入/输出井组成的。

3.4.4　磁盘调度

磁盘调度分为移臂调度和旋转调度两类，并且是先进行移臂调度，然后再进行旋转调度。由于访问磁盘最耗时的是寻道时间，因此，磁盘调度的目标是使磁盘的平均寻道时间最少。

1. 磁盘驱动调度

一般可采用以下四种磁盘调度算法。

(1) 先来先服务磁盘调度算法(FCFS)。这是最简单的磁盘调度算法。它根据进程请求访问磁盘的先后次序进行调度。优点是公平、简单，且每个进程的请求都能依次得到处理，不会出现某个进程的请求长期得不到满足的情况。此算法由于未对寻道进行优化，因此平均寻道时间可能较长。

(2) 最短寻道时间优先磁盘调度算法(SSTF)。SSTF 算法要求访问的磁道与当前磁头所在的磁道距离最近，使得每次的寻道时间最短，但这种调度算法却不能保证平均寻道时间最短。

(3) 扫描算法(SCAN)。SCAN 算法也是一种寻道优化的算法，它克服了 SSTF 算法的缺点。既考虑了访问磁道与磁头当前位置的距离，更优先考虑了当前的移动方向。这种算法磁头移动的规律颇似电梯的运行，故又常称为电梯调度算法。

(4) 单向扫描调度算法(CSCAN)。SCAN 存在这样的问题：当磁头刚从里向外移动过某一磁道时，若恰有一进程请求访问此磁道，则该进程必须等待磁头从里向外，然后再从外向里扫描完所有要访问的磁道后，才能处理该进程的请求，致使该进程的请求被严重推迟。为了减少这种延迟，CSCAN 算法规定了磁头做单向移动。

2. 旋转调度算法

系统应该选择延迟时间最短的进程对磁盘的扇区进行访问。当有若干等待进程请求访问磁盘上的信息时，旋转调度应考虑以下三种情况。

(1) 进程请求访问的是同一磁道上的不同编号的扇区。

(2) 进程请求访问的是不同磁道上的不同编号的扇区。

(3) 进程请求访问的是不同磁道上具有相同编号的扇区。

真题链接

【例 3-8】假设磁盘块与缓冲区大小相同，每个盘块读入缓冲区的时间为 10μs，由缓冲区送至用户区的时间是 5μs，系统对每个磁盘块的处理时间为 2μs，若用户需要将大小为 10 个磁盘块的 Doc 文件逐块从磁盘读入缓冲区，并送至用户区进行处理，那么采用单缓冲区需要花费的时间为 (25) μs，采用双缓冲区花费的时间为 (26) μs (2014 年 11 月真题 25、26)

A. 100　　B. 107　　C. 152　　D. 170

【解析】单缓冲区：(10+5)×10+2=152

双缓冲区：10×10+5+2=107

【答案】(25)C；(26)B

3.5 文 件 管 理

考核说明：本节主要介绍文件的基本概念、结构和组织，文件目录，文件的存取方法和空间管理，文件的使用及安全管理。其中，文件的目录结构、文件存储空间的管理是常考考点，需要掌握。

3.5.1 文件与文件系统

1. 文件

文件是具有符号名的、在逻辑上具有完整意义的一组相关信息项的集合。信息项是构成文件内容的基本单位，可以是一个字符，也可以是一个记录，记录可以等长，也可以不等长。一个文件包括文件体和文件说明。

2. 文件系统

所谓文件管理系统就是操作系统中实现文件统一管理的一组软件和相关数据的集合。专门负责管理和存取文件信息的软件机构，简称文件系统。文件系统的功能包括：按名存取、统一的用户接口、并发访问和控制、安全性控制、优化性能、差错恢复。

3. 文件类型

可以按不同的标准对文件进行分类。

- 按文件的性质和用途分类，可以分为系统文件、库文件和用户文件。

- 按信息保存期限分类，可以分为临时文件、档案文件和永久文件。
- 按文件的保护方式分类，可以分为只读文件、读写文件、可执行文件和不保护文件。
- UNIX 系统将文件分为普通文件、目录文件和特殊文件。
- 目前常用的文件系统类型有 FAT、VFAT、NTFS、EXT2、HPFS 等。

文件分类的目的是对不同文件进行管理，以提高系统效率和用户界面的友好性。

3.5.2　文件的结构和组织

文件的结构是指文件的组织形式，从用户观点所看到的文件组织形式，称为文件的逻辑结构；从实现观点考察文件在辅助存储器上的存放方式，常称为文件的物理结构。

1. 文件的逻辑结构

文件的逻辑结构可以分为两类：一类是有结构的记录式文件，它是由一个以上的记录构成的文件；另一类是无结构的流式文件，它是由一串顺序字符流构成的文件。

(1) 有结构的记录式文件。记录文件根据长度可分为定长和不定长两种。

(2) 无结构的流式文件。无结构的流式文件通常采用顺序访问方式，并且每次读写访问可以指定任意数据长度，其长度以字节为单位。

2. 文件的物理结构

文件的物理结构是指文件的内部组织形式，也就是文件在物理存储设备上的存放方法。常用的文件物理结构有以下三种。

(1) 连续结构。连续结构也称顺序结构。这是一种最简单的物理结构，它把逻辑上连续的文件信息依次存放在连续编号的物理块中。只要知道文件在存储设备上的起始地址(首块号)和文件长度(总块数)，就能很快地进行存取。

(2) 链接结构。链接结构也叫串联结构。它是将逻辑上连续的文件信息存放在不连续的物理块中，每个物理块设有一个指针指向下一个物理块。只要指明文件的第一个物理块号，就可以利用链指针检索整个文件。

(3) 索引结构。采用索引结构将逻辑上连续的文件信息存放在不连续的物理块中，系统为每个文件建立一张索引表。索引表记录了文件信息所在的逻辑块号对应的物理块号，并将索引表的起始地址放在文件对应的文件目录项中。

UNIX 文件系统采用的是三级索引结构，文件系统中的 inode 是基本的构件，它表示文件系统树型结构的结点。UNIX 有直接、一级间接、二级间接、三级间接四种寻址方式。

3.5.3　文件目录

系统为每个文件设置一个描述性数据结构——文件控制块(File Control Block，FCB)，文件目录就是文件控制块的有序集合。

1. 文件控制块

文件控制块(FCB)是系统为管理文件而设置的一个数据结构。FCB 是文件存在的标志，它记录了系统管理文件所需要的全部信息。FCB 通常应包括三类信息。

(1) 基本信息类：文件名、文件的物理位置、文件长度、文件块数等。

(2) 存取控制信息类：文件的存取权限。

(3) 使用信息类：文件的建立日期、最后一次修改的日期、最后一次访问的日期，当前使用的信息和目录文件。

2. 目录结构

文件目录结构的组织方式会直接影响文件的存取速度，关系到文件共享性和安全性。常见的目录结构有三种：一级目录结构、二级目录结构和多级目录结构。目前大多数操作系统(如 UNIX、DOS 等)都采用多级目录结构，又称树型目录结构。

1) 一级目录结构

一级目录的整个目录组织是一个线性结构，在整个系统中只需建立一张目录表，系统为每个文件分配一个目录项(文件控制块)。它主要用在单用户环境中。

2) 二级目录结构

二级目录结构是由主文件目录(Master File Directory，MFD)和用户目录(User File Directory，UFD)组成的。在主文件目录中，每个用户文件目录都占有一个目录项，其目录项中包括用户名和指向该用户目录文件的指针。用户目录由用户的所有文件的目录项组成。

3) 多级目录结构

在多道程序设计系统中常采用多级目录结构，这种目录结构就像一棵倒置的有根树，所以也称为树型目录结构。从树根向下，每一个结点是一个目录，叶结点是文件。MS-DOS 和 UNIX 等操作系统均采用多级目录结构。

真题链接

【例 3-9】若某文件系统的目录结构如图 3-9 所示，假设用户要访问文件 fault.swf，且当前工作目录为 swshare，则该文件的全文件名为__(27)__，相对路径和绝对路径分别为__(28)__。(2014 年 5 月真题 27～28)

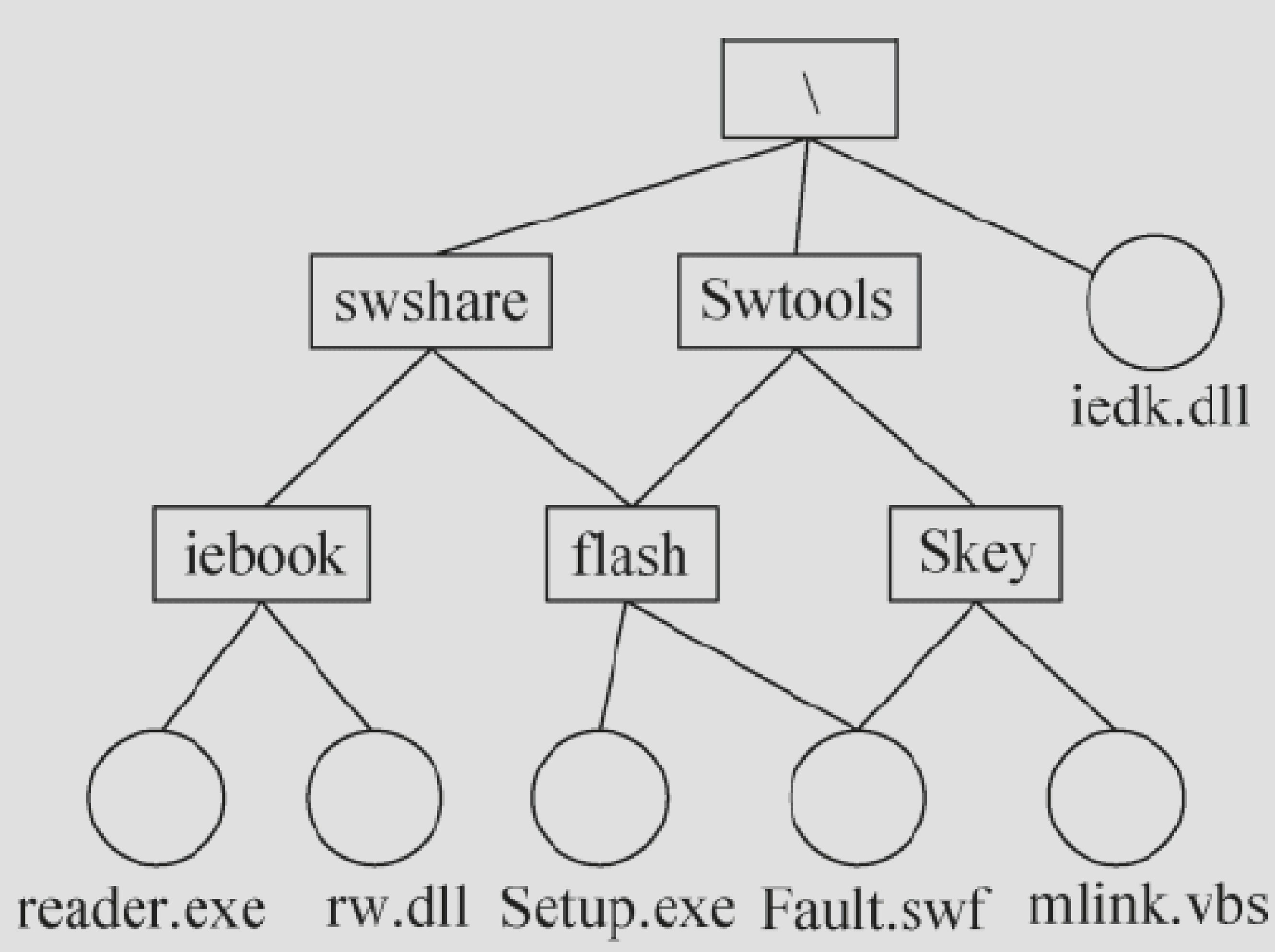


图 3-9 文件系统的目录结构

(27) A.fault.swf　B.flash\fault.swf　C.swshare\flash\fault.swf　D.\swshare\flash\fault.swf

(28) A.swshare\flash\和\flash\　　　　B. flash\和\swshare\flash\
C.\swshare\flash\和 flash\　　　　D. \flash\和\swshare\flash\

【解析】全文件名应该从根目录开始，因此为\swshare\flash\fault.swf。相对路径是从当前路径开始的路径，fault.swf 在当前工作目录 swshare 下的 flash 文件夹中，因此相对路径为 flash\。绝对路径是指从根目录开始的路径，即\swshare\flash\

【答案】(27)D；(28)B

3.5.4　存取方法和存取空间的管理

1. 文件的存取方法

文件的存取方法是指读写文件存储器上的一个物理块的方法。通常有顺序存取、随机存取和按键存取等。

(1) 顺序存取，就是按从前到后的次序依次访问文件的各个信息项。

(2) 随机存取，又称直接存取，即允许用户随意存取文件的任意一个物理记录。

(3) 按键存取，即根据文件中各记录的某个数据项内容来存取记录，这种数据项称为“键”。因此，将这种存取法称为按键存取。

2. 文件存储空间的管理

外存空间管理的数据结构通常称为磁盘分配表。常用的空间管理方法有空闲区表、位示图、空闲块链和成组链接法四种。

(1) 空闲区表。将外存空间上一个连续未分配区域称为空闲区。操作系统为磁盘外存上所有空闲区建立一张空闲表，每个表项对应一个空闲区，空闲表中包含序号、空闲区的第一块号、空闲块的块数等信息。它适用于连续文件结构。

(2) 位示图。在外存上建立一张位示图，记录文件存储器的使用情况。每一位对应文件存储器上的一个物理块，取值 0 和 1 分别表示空闲和占用。这种方法的主要特点是位示图的大小由磁盘空间的大小(物理块总数)决定，位示图的描述能力强，适合各种物理结构。

(3) 空闲块链。每个空闲物理块中有指向下一个空闲物理块的指针，所有空闲物理块构成一个链表，链表的头指针放在文件存储器的特定位置上(如管理块中)。

(4) 成组链接法。在 UNIX 系统中，将空闲块分成若干组，每 100 个空闲块为一组，每组的第一个空闲块登记了下一组空闲块的物理盘块号和空闲块总数，如果一个组的第一个空闲块号为 0，则意味着该组是最后一组，即无下一组空闲块。

3.5.5　文件的使用

操作系统在操作级(命令级)和编程级(系统调用和函数)向用户提供文件的服务。操作系统在操作级向用户提供的命令有目录管理类命令、文件操作类命令(如复制、删除和修改)、文件管理类命令(如设置文件权限)等。操作系统在编程级向用户提供的系统调用主要有以下六种。

- 创建文件：如 create(文件名，参数表)。
- 删除文件：如 delete(文件名)。
- 打开文件：如 open(文件名，参数表)。
- 关闭文件：如 close(文件名)。
- 读文件：如 read(文件名，参数表)。
- 写文件：如 write(文件名，参数表)。

3.5.6 文件的共享和保护

1. 文件的共享

文件共享是指不同的用户进程使用同一文件。文件共享有多种形式，采用文件名和文件说明分离的目录结构有利于实现文件共享。常见的文件链接有硬链接和符号链接两种。

2. 文件的保护

文件系统对文件的保护常采用存取控制方式进行。所谓存取控制就是不同的用户对文件的访问有不同的权限，以防止文件被未经文件主同意的用户访问。

1) 存取控制矩阵

理论上，存取控制可用存取控制矩阵方法，它是一个二维矩阵，一维列出计算机的全部用户，另一维列出系统中的全部文件。

2) 存取控制表

存取控制表是按用户对文件访问权限的差别对用户进行分类，由于某一文件往往只与少数几个用户有关，所以这种分类方法可使存取控制表大为简化。

3) 用户权限表

用户权限表是以用户或用户组为单位将用户可存取的文件集中起来存入表中，表中每个表目表示该用户对相应文件的存取权限。

4) 密码

在创建文件时，由用户提供一个密码，在文件存入磁盘时用该密码对文件内容加密。进行读取操作时，要对文件进行解密，只有知道密码的用户才能读取文件。

3.5.7 系统的安全

一般从四个级别上对文件进行安全性管理：系统级、用户级、目录级和文件级。

(1) 系统级安全管理的主要任务是不允许未经授权的用户进入系统，从而也防止了他人非法使用系统中的各类资源(包括文件)。系统级管理的主要措施有注册与登录。

(2) 用户级安全管理是通过对所有用户分类和为指定用户分配访问权限，不同的用户通过对不同文件设置不同的存取权限来实现。有的系统将用户分为超级用户、系统操作员和一般用户。

(3) 目录级安全管理是为了保护系统中各种目录而设计的，与用户权限无关。为保证

目录的安全，规定只有系统核心才具有写目录的权利。

(4) 文件级安全管理是通过系统管理员或文件主对文件属性的设置来控制用户对文件的访问。通常可设置的属性有只执行、隐含、只读、读写、共享、系统。

3.6 应试加油站

3.6.1 考频统计

通过对近几年考试真题的分析，“操作系统”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 3-1 所示。

表 3-1　历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：23～28	操作系统概念、死锁、段页式存储系统、文件系统	6
	下午：无		0
2014 年 5 月	上午：23～28	PV 操作、段页式存储系统、缓冲技术、进程状态图	6
	下午：无		0
2013 年 11 月	上午：23～28	PV 操作、段页式存储管理、文件管理、互斥资源分配	6
	下午：无		0
2013 年 5 月	上午：23～28	时间片轮转法、线程、进程状态转换、分区存储管理	6
	下午：无		0
2012 年 11 月	上午：23～28	避免死锁、作业调度、文件索引	6
	下午：无		0
2012 年 5 月	上午：23～28	银行家算法、虚拟页式存储管理	
	下午：无		0

纵观历年试卷，本章知识点是以选择题和综合分析题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量大约为 6 道选择题，所占分值为 6 分(约占试卷总分值 75 分中的 8%)。本章主要考查相关的理论知识，难度中等偏低，进程控制、各种存储管理方法、磁盘调度、文件目录及文件的存取方法和存储空间的管理是重点；进程的 PV 操作和死锁的产生原因及处理稍微偏难，要正确理解和掌握。

3.6.2 解题技巧

【典型题 3-1】假设某分时系统采用简单时间片轮转法，当系统中的用户数为 n、时间片为 q 时，系统对每个用户的响应时间 T=__(23)__。(2013 年 5 月真题 23)

(23) A. n　　B. q　　C. n×q　　D. n+q

【解析】在时间片轮转法中，系统将所有的就绪进程按先来先服务的原则排成一个队列，每次调度时，

把 CPU 分配给队首进程，并令其执行一个时间片。当执行的时间片用完时，中断请求，将该程序送往就绪队列的队尾，并把处理机分配给新的队首进程，同时让它也执行一个时间片。这样就保证了就绪队列中的所有进程在一给定的时间内均能获得一个时间片的处理机执行时间。按照这样的思想，每个用户就是一个进程，这样每个用户的响应时间为 n×q。

【答案】 C

【典型题 3-2】 在支持多线程的操作系统中，假设进程 P 创建了若干个线程，那么 (24) 是不能被这些线程共享的。(2013 年 5 月真题 24)

(24) A. 该进程的代码段　　B. 该进程中打开的文件

C. 该进程的全局变量　　D. 该进程中某线程的栈指针

【解析】 线程共享的环境包括：进程代码段、进程的公有数据(利用这些共享的数据，线程很容易地实现相互之间的通信)、进程打开的文件描述符、信号的处理器、进程的当前目录和进程用户 ID 与进程组 ID(选项 C，即进程的全局变量)。但该进程中某线程的栈指针不能被这些线程所共享。

【答案】 D

【典型题 3-3】 进程资源图如图 3-10 中(a)和(b)所示，其中：图(a)中 (25) ；图(b)中 (26) 。(2013 年 5 月真题 25～26)

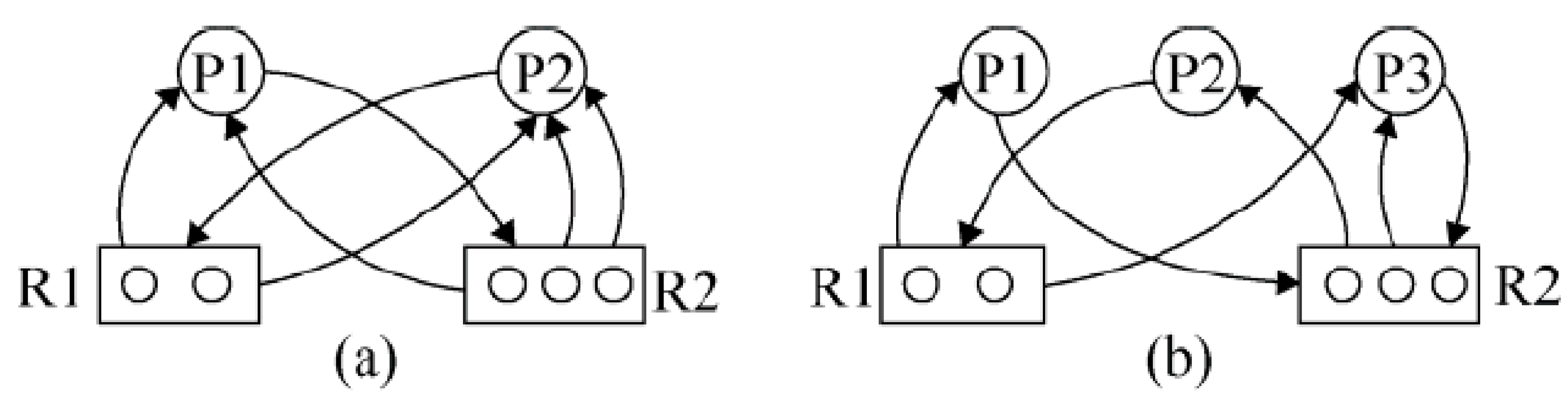


图 3-10　进程资源图

(25) A. P1 是非阻塞节点，P2 是阻塞节点，所以该图不可以化简、是死锁的

B. P1、P2 都是阻塞节点，所以该图不可以化简、是死锁的

C. P1、P2 都是非阻塞节点，所以该图可以化简、是非死锁的

D . P1 是阻塞节点，P2 是非阻塞节点，所以该图不可以化简、是死锁的

(26) A. P1、P2、P3 都是非阻塞节点，该图可以化简、是非死锁的

B. P1、P2、P3 都是阻塞节点，该图不可以化简、是死锁的

C. P2 是阻塞节点，P1、P3 是非阻塞节点，该图可以化简、是非死锁的

D. P1、P2 是非阻塞节点，P3 是阻塞节点，该图不可以化简、是死锁的

【解析】 由(a)图可知，R1 资源共有两个，都被进程 P1 和 P2 所占用，R2 资源有三个，都被进程 P1、P2 所占用，而此时进程 P1 在请求 R2 资源，P2 在请求 R1 资源，而 R1 和 R2 已经没有资源可用了，导致两个进程都进入了死锁状态。

由(b)图可知，R1 的两个资源已经分别被 P1 和 P3 所占用，R2 的三个资源已经占用了两个，当进程 P1 和 P3 请求资源 R2 的时候，无论分配给任何一方都可以使两个进程满足所需的资源，从而可以化简，使得 P2 也能满足对资源 R1 的需求。因此 P1 和 P3 是非阻塞节点，P2 是阻塞节点，但最终可以化简。

【答案】 (25)B；(26)C

【典型题 3-4】 假设系统采用 PV 操作实现进程同步与互斥，若有 n 个进程共享一台扫描仪，那么当信号量 S 的值为−3 时，表示系统中有 (23) 个进程等待使用扫描仪。(2013 年 11 月真题 23)

(23) A. 0　　B. $n-3$　　C. 3　　D. n

【解析】本题考查操作系统的设备管理。PV 操作与信号量的处理相关，P 表示通过的意思，V 表示释放的意思。执行 P 操作时信号量 S 的值减 1，若结果大于等于 0，则 P(S)执行完毕，否则执行 P 操作的进程暂停以等待释放，S 为−3 时，有 3 个进程等待扫描。

【答案】C

【典型题 3-5】假设系统中有三类互斥资源 R1、R2 和 R3，可用资源数分别为 10、5 和 3。在 T_0 时刻系统中有 P1、P2、P3、P4 和 P5 五个进程，这些进程对资源的最大需求量和已分配资源数如表 3-2 所示，此时系统剩余的可用资源数分别为 (27) 。如果进程按 (28) 序列执行，那么系统状态是安全的。(2013 年 11 月真题 27～28)

表 3-2　某些进程对资源的最大需求量和已分配资源数

资源 / 进程	最大需求量			已分配资源数		
	R1	R2	R3	R1	R2	R3
P1	5	3	1	1	1	1
P2	3	2	0	2	1	0
P3	6	1	1	3	1	0
P4	3	3	2	1	1	1
P5	2	1	1	1	1	0

(27) A. 1、1 和 0　　B. 1、1 和 1　　C. 2、1 和 0　　D. 2、0 和 1

(28) A. P1→P2→P4→P5→P3　　B. P5→P2→P4→P3→P1

C. P4→P2→P1→P5→P3　　D. P5→P1→P4→P2→P3

【解析】可用资源数减去已分配资源数就是可用资源数，所以空(27)处的答案选 D。对于空(28)，由于 R1、R2、R3 可用资源数为 2、0、1，如果先执行 P1，需要 R1 资源数为 5−1=4>2，所以选项 A 错；如果先执行 P5，还需要 R1、R3 的资源数分别为 1、1，可以满足，执行完 P5，会将已分配的资源数还给 R1、R2、R3，此时 R1、R2、R3 可用资源数为 3、1、1，此时若再执行 P1，仍然会出现死锁，即 R1 所需资源数 5−1=4>3，所以选项 D 错，同理分析先执行 P4 时需要 R2 的资源数 3−1=2>0，所以答案 C 错。

【答案】(27)D；(28)B

【典型题 3-6】某系统中仅有 5 个并发进程竞争某类资源，且都需要 3 个该类资源，那么至少有 (23) 个该类资源，才能保证系统不会发生死锁。(2012 年 11 月真题 23)

(23) A. 9　　B. 10　　C. 11　　D. 15

【解析】要保证不会发生死锁，至少有一个进程能申请到 3 个资源，其他进程都申请到 2 个资源，这时需要 11 个资源。当占用 3 个资源的进程完成后，释放资源，其他进程便可以申请到更多的资源。如果只有 10 个资源，可能每个进程申请到 2 个资源，此时便发生了死锁。

【答案】C

【典型题 3-7】若某企业拥有的总资金数为 15，投资 4 个项目 P1、P2、P3、P4，各项目需要的最大资金数分别是 6、8、8、10，企业资金情况如图 3-11(a)所示。P1 新申请 2 个资金，P2 新申请 1 个资金，若企业资金管理处为项目 P1 和 P2 分配新申请的资金，则 P1、P2、P3、P4 尚需的资金数分别为 (23) ，假设 P1 已经还清所有投资款，企业资金使用情况如图 3-11(b)所示，那么企业的可用资金数为 (24) 。若在图 3-11(b)所示的情况下，企业资金管理处为 P2、P3、P4 各分配资金数 2、2、3，则分配后 P2、P3、P4 已用资金数分别为 (25) 。(2012 年 5 月真题 23～25)

项目	最大资金	已用资金	尚需资金
P1	6	2	4
P2	8	3	5
P3	8	2	6
P4	10	3	7

(a)

项目	最大资金	已用资金	尚需资金
P1	—	—	—
P2	8	3	5
P3	8	2	6
P4	10	3	7

(b)

图 3-11　企业资金情况

(23) A. 1、3、6、7，可用资金数为 0，故资金周转状态是不安全的

B. 2、5、6、7，可用资金数为 1，故资金周转状态是不安全的

C. 2、4、6、7，可用资金数为 2，故资金周转状态是安全的

D. 3、3、6、7，可用资金数为 2，故资金周转状态是安全的

(24) A. 4　　B. 5　　C. 6　　D. 7

(25) A. 3、2、3，尚需资金数分别为 5、6、7，故资金周转状态是安全的

B. 5、4、6，尚需资金数分别为 3、4、4，故资金周转状态是安全的

C. 3、2、3，尚需资金数分别为 5、6、7，故资金周转状态是不安全的

D. 5、4、6，尚需资金数分别为 3、4、4，故资金周转状态是不安全的

【解析】由图(a)可知，之前已经给 4 个项目分配了 2+3+2+3=10 个资金。P1 新申请了 2 个资金，P2 新申请 1 个资金，则可用资金为 15−10−2−1=2，P1、P2、P3、P4 已用的资金数分别为 4、4(3+1)、2、3，尚需的资金数为 2(6−4)、4(8−4)、6、7。故空(23)选 C。

由图(b)可知，已用资金数为 3+2+3=8，可用资金数为 15−8=7，故空(24)选 D。

在图(b)所示的情况下，企业资金管理处为 P2、P3、P4 各分配资金数 2、2、3，则 P2、P3、P4 已用的资金数分别为 3+2、2+2、3+3，即 5、4、6。尚需资金为 3、4、4，而此时总资金数为 0，故资金周转状态是不安全的。因此空(25)选 D。

【答案】(23)C；(24)D；(25)D

【典型题 3-8】假设段页式存储管理系统中的地址结构如图 3-12 所示，则系统中 (24) 。(2013 年 11 月真题 24)

31　　　　24	23　　　　12	11　　　　0
段　号	页　号	页内地址

图 3-12　某段页式存储管理系统中的地址结构

(24) A. 页的大小为 4K，每个段的大小均为 4096 个页，最多可有 256 个段

B. 页的大小为 4K，每个段最大允许有 4096 个页，最多可有 256 个段

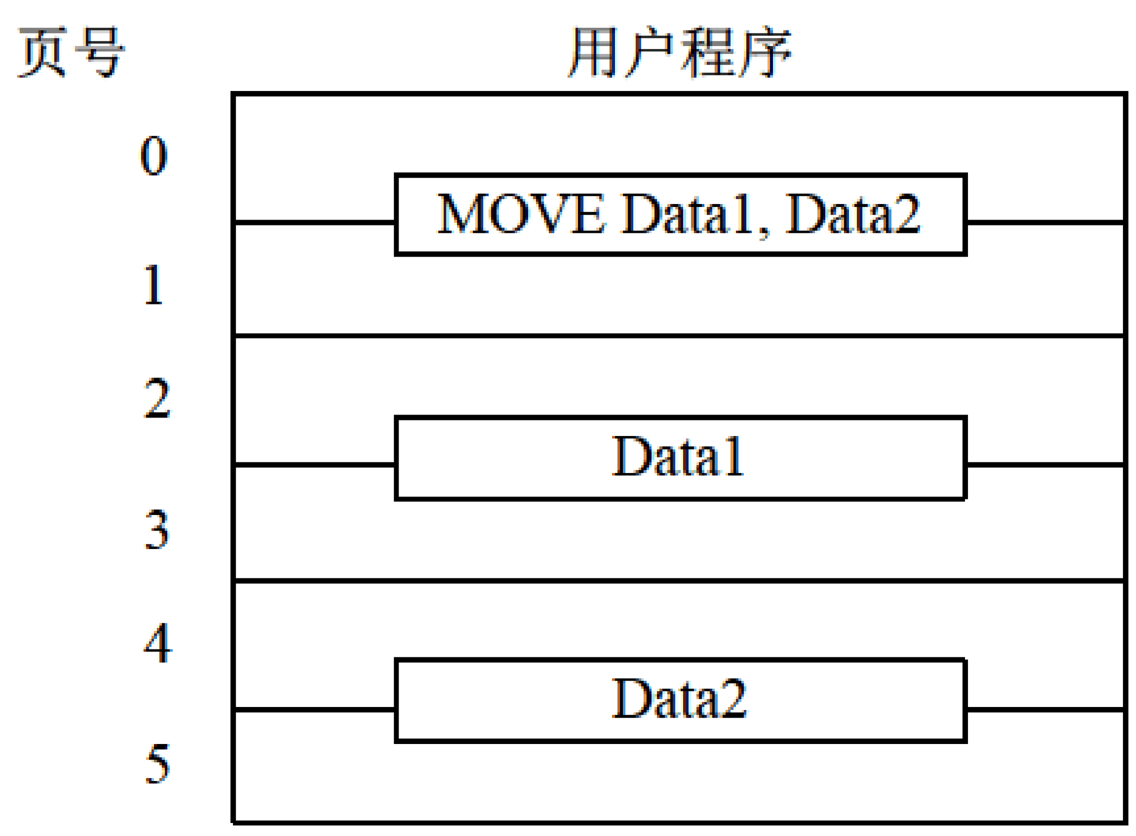
C. 页的大小为 8K，每个段的大小均为 2048 个页，最多可有 128 个段

D. 页的大小为 8K，每个段最大允许有 2048 个页，最多可有 128 个段

【解析】页内地址位数是 12 位，大小为 4K，页号地址为 12～24 位，最多可以容纳 4096 个页。段号地址为 8 位，最多可以容纳 256 个段。

【答案】B

【典型题 3-9】假设一按字节编址的 16 位计算机系统，采用虚拟页式存储管理方案，页面的大小为 2K，且系统中没有使用快表(或联想存储器)。某用户程序如图 3-13(a)所示，该程序的页面变换表如图 3-13(b)所示，表中状态位 1 和 0 分别表示页面在内存或不在内存。



(a)

页面变换表

页号	状态
0	1
1	0
2	0
3	0
4	0
5	0

(b)

图 3-13　用户程序及对应的页面变换表

图(a)中的 MOVE Data1，Data2 是一个 4 字节的指令，Data1 和 Data2 表示该指令的两个 32 位操作数。假设 MOVE 指令存放在 2047 地址开始的内存单元中，Data1 存放在 6143 地址开始的内存单元中，Data2 存放在 10239 地址开始的内存单元中，那么执行 MOVE 指令将产生<u>　(26)　</u>次缺页中断，其中取指令产生<u>　(27)　</u>次缺页中断，取 Data1 和 Data2 操作数分别产生<u>　(28)　</u>次缺页中断。(2012 年 5 月真题 26～28)

(26) A. 3　　B. 4　　C. 5　　D. 6

(27) A. 0　　B. 1　　C. 2　　D. 3

(28) A. 1、1　　B. 1、2　　C. 2、2　　D. 2、3

【解析】由图(a)可知，MOVE 指令、Data1、Data2 分别占 2 个页号，由页面变换表可知，只有页号为 0 的页面在内存中，不需要产生缺页中断，取指的时候产生 1 次缺页中断，取操作数的时候分别产生 2 次缺页中断，总的缺页中断次数为 5。

【答案】(26)C；(27)B；(28)C

【典型题 3-10】某文件管理系统采用位示图(bitmap)记录磁盘的使用情况。如果系统的字长为 32 位，磁盘物理块的大小为 4 MB，物理块依次编号为 0、1、2、…，位示图字依次编号为 0、1、2、…，那么 16385 号物理块的使用情况在位示图中的第<u>　(25)　</u>个字中描述；如果磁盘的容量为 1000GB，那么位示图需要<u>　(26)　</u>个字来表示。(2013 年 11 月真题 25～26)

(25) A. 128　　B. 256　　C. 512　　D. 1024

(26) A. 1200　　B. 3200　　C. 6400　　D. 8000

【解析】16385÷32=512，所以可以在第 512 个字描述；1000×1024÷4÷32=8000，位图需要 8000 个字来表示。

【答案】(25)C；(26)D

【典型题 3-11】设文件索引节点中有 8 个地址项，每个地址项大小为 4 字节，其中 5 个地址项为直接地址索引，2 个地址项为一级间接索引，1 个地址项为二级间接索引。磁盘索引块和磁盘数据块大小为 1KB。若要访问文件的逻辑块号分别为 8 和 518，则系统应分别采用<u>　(27)　</u>，而且可以表示的单个文件最大长度为<u>(28)</u>KB。(2012 年 11 月真题 27、28)

(27) A. 直接地址索引和一级间接地址索引

B. 直接地址索引和二级间接地址索引

C. 一级间接地址索引和二级间接地址索引

D. 一级间接地址索引和一级间接地址索引

(28) A. 517　　B. 1029　　C. 16 513　　D. 66 053

【解析】若逻辑块号<5，则为直接块，使用直接地址索引；若 5≤逻辑块号<261(2^8+5)，则采用一级间接地址索引；若 261≤逻辑块号<65 797(2^{16}+261)，则采用二级间接地址索引。因此若要访问文件的逻辑块号分别为 8 和 518，则系统应分别采用一级间接地址索引和二级间接地址索引。

磁盘索引块的大小为 1KB，每个盘块号为 4 字节，则一个磁盘块可以保存的盘块号的个数为 1K÷4=256。

因为每个磁盘索引块和磁盘数据块大小均为 1KB，所以 5 个直接地址索引指向的数据块大小为 5KB，2 个一级间接索引共包括 2×256 个直接地址索引，其指向的数据块大小为 2×256×1KB=512KB，1 个二级间接索引所包含的直接地址索引为 256×256×1KB=65 536KB。8 个地址项所指向的数据块总大小即单个文件最大长度为(65 536+512+5)KB，即 66 053KB。

【答案】(27)C；(28)D

【典型题 3-12】某计算机系统中有一个 CPU、一台输入设备和一台输出设备，假设系统中有三个作业 T1、T2 和 T3，系统采用优先级调度，且 T1 的优先级>T2 的优先级>T3 的优先级。若每个作业具有三个程序段：输入 I_i、计算 C_i 和输出 P_i(i=1,2,3)，执行顺序为 I_i、C_i、P_i，则这三个作业各程序段并发执行的前驱图如图 3-14 所示。图中①、②分别为<u>(24)</u>，③、④分别为<u>(25)</u>，⑤、⑥分别为<u>(26)</u>。(2012 年 11 月真题 24～26)

(24) A. I_2、C_2　　B. I_2、I_3　　C. C_1、P_3　　D. C_2、P_2

(25) A. C_1、C_2　　B. I_2、C_1　　C. I_3、P_3　　D. C_1、P_2

(26) A. I_3、C_2　　B. I_2、C_1　　C. P_2、P_3　　D. C_1、P_2

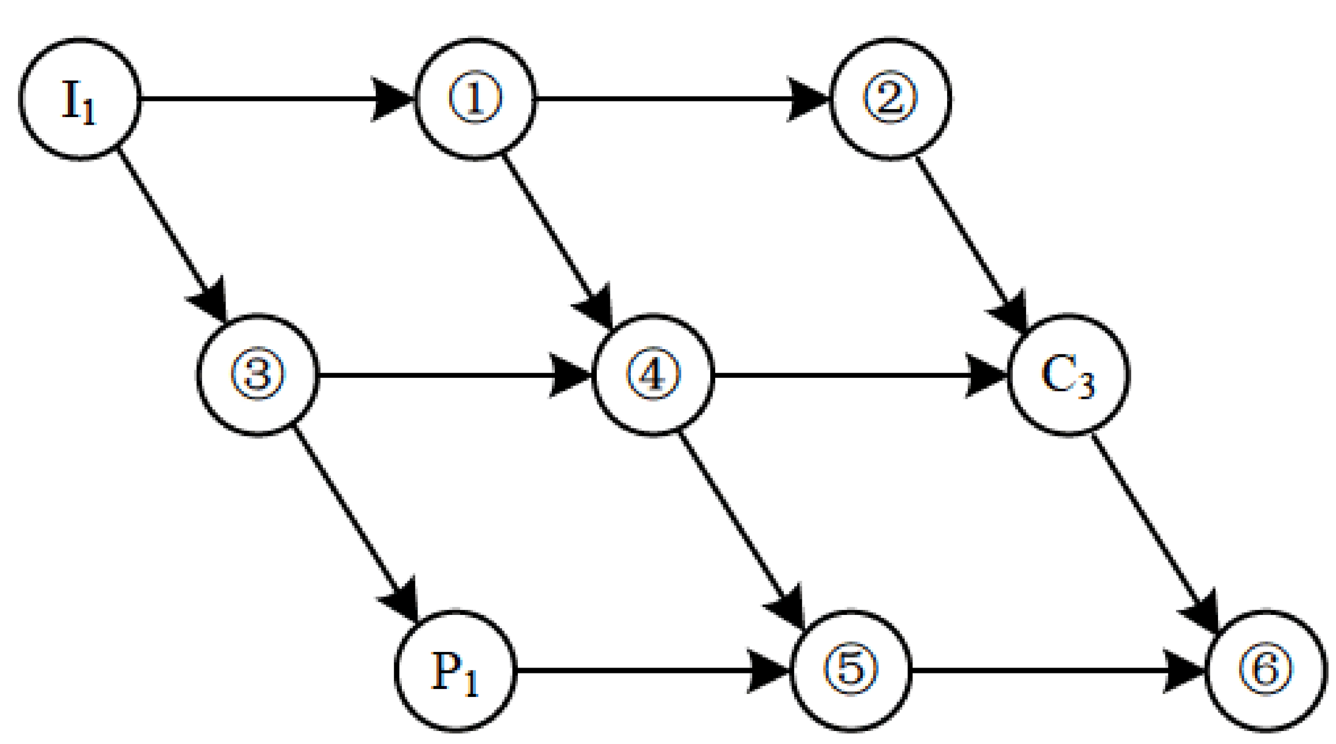


图 3-14　并发执行的前驱图

【解析】T1 的优先级最高，则首先从作业 T1 开始执行。当 T1 的输入完成后，便可进行计算；同时输入设备空闲，可以执行作业 T2 的输入操作，由此可以判断①和③中一个是 I_2 一个是 C_1。很显然，P_1 的前驱是 C_1，因为只有 C_1 执行完后，输入设备才会空闲，输入设备空闲后，才能执行 P_1。由此得知③为 C_1，则①为 I_2。

而以 C_1 为前驱的，一是 P_1，二是 C_2，故④为 C_2。T1 的计算完成后，便可进行输出操作，同时计算设备空闲，可以进行 T2 的计算操作，很显然，②为 I_3。

以此类推，得知⑤、⑥为 P_2、P_3。

【答案】(24)B；(25)A；(26)C

3.7　过 关 习 题

1. 某企业生产流水线 M 共有两位生产者，生产者甲不断地将其工序上加工的半成品放入半成品箱，生产者乙从半成品箱取出继续加工。假设半成品箱可存放 n 件半成品，采用 PV 操作实现生产者甲和生产者乙的同步可以设置三个信号量 S、S1 和 S2，其同步模型如图 3-15 所示。

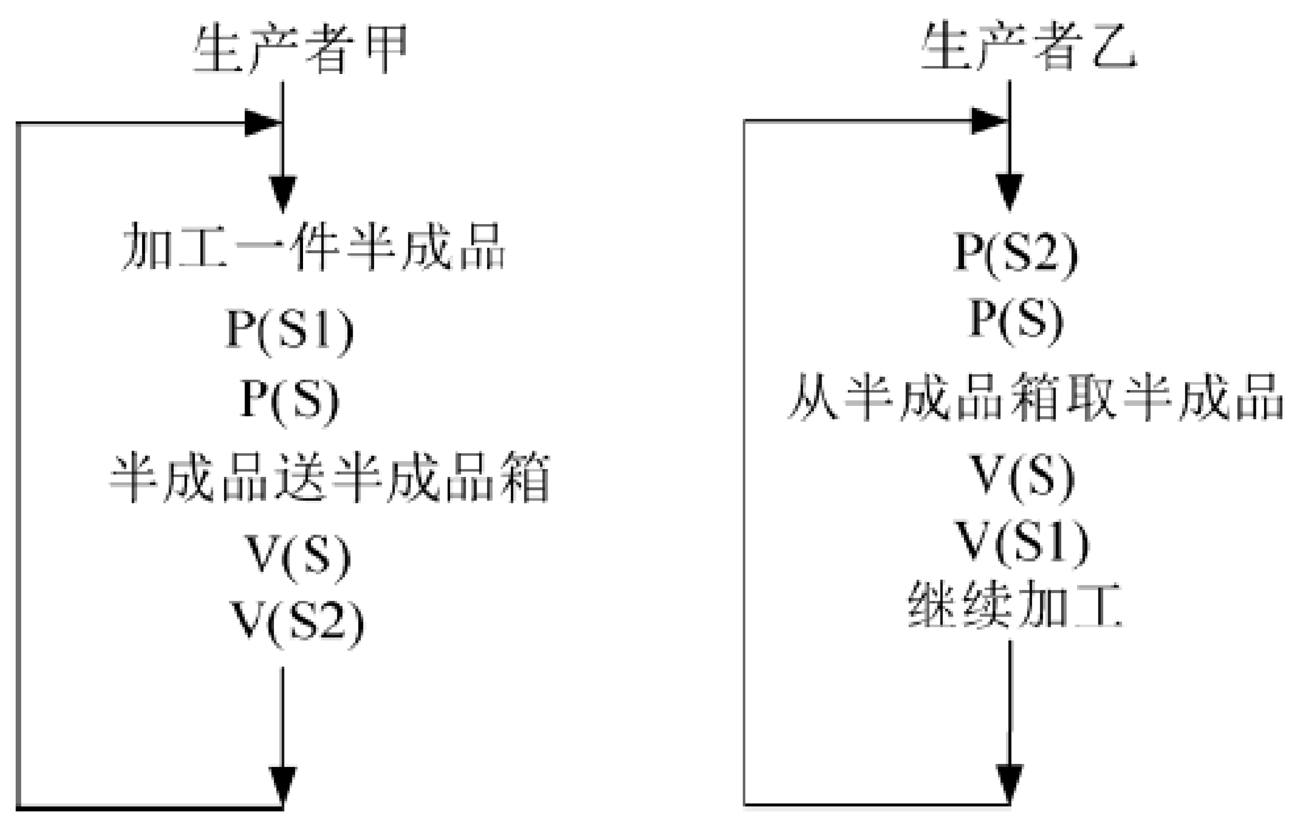


图 3-15　同步模型

信号量 S 是一个互斥信号量，初值为__(1)__；S1、S2 的初值分别为__(2)__。

(1)　A. 0　　B. 1　　C. n　　D. 任意正整数

(2)　A. n、0　　B. 0、n　　C. 1、n　　D. n、1

2. 进程 P1、P2、P3、P4 和 P5 的前驱图如图 3-16 所示。

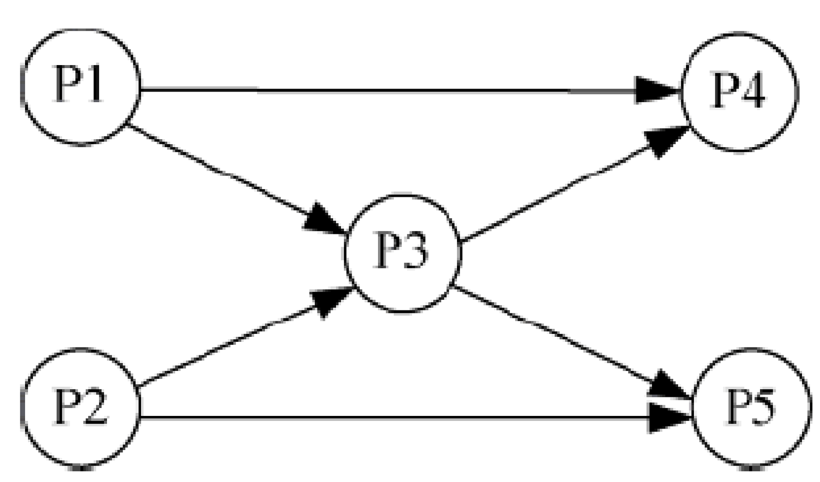


图 3-16　前驱图

若用 PV 操作控制进程 P1～P5 并发执行的过程，则需要设置 6 个信号 S1、S2、S3、S4、S5 和 S6，且信号量 S1～S6 的初值都等于零。图 3-17 中 a 和 b 处应分别填写__(1)__；c 和 d 处应分别填写__(2)__，e 和 f 处应分别填写__(3)__。

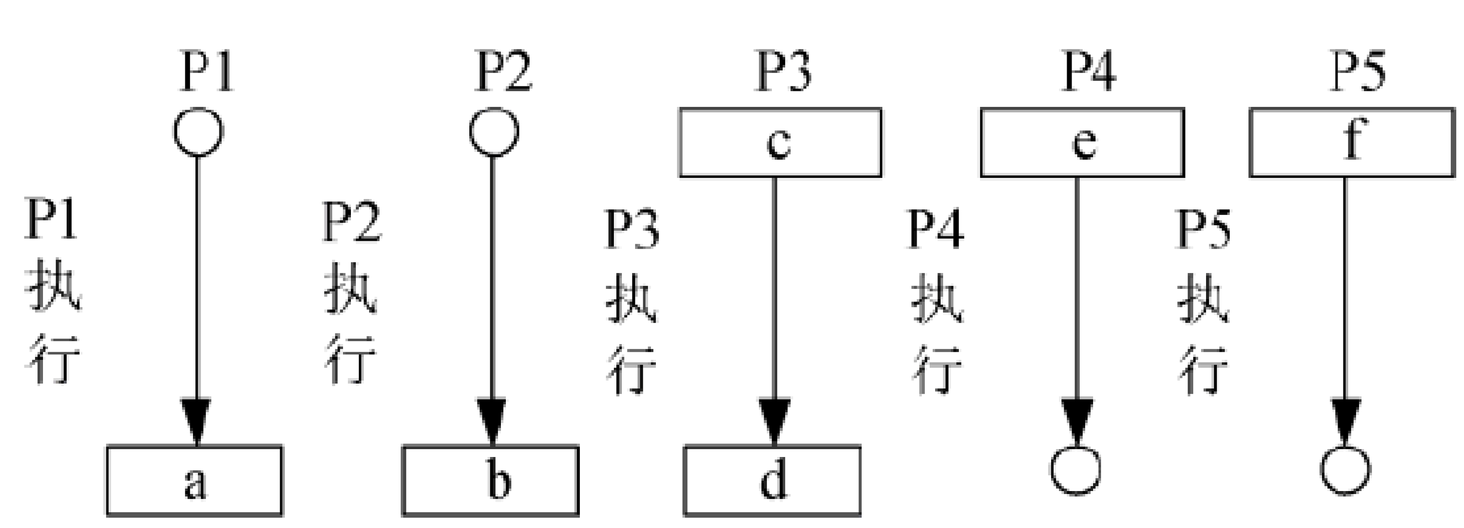


图 3-17　进程的执行过程

(1)　A. P(S1)P(S2)和 P(S3)P(S4)　　B. P(S1)V(S2)和 P(S2)V(S1)

C. V(S1)V(S2)和 V(S3)V(S4)　　D. P(S1)P(S2)和 V(S1)V(S2)

(2) A. P(S1)P(S2)和 V(S3)V(S4)　　B. P(S1)P(S3)和 V(S5)V(S6)

C. V(S1)V(S2)和 P(S3)P(S4)　　D. P(S1)V(S3)和 P(S2)V(S4)

(3) A. P(S3)P(S4)和 V(S5)V(S6)　　B. V(S5)V(S6)和 P(S5)P(S6)

C. P(S2)P(S5)和 P(S4)P(S6)　　D. P(S4)V(S5)和 P(S5)V(S6)

3. 系统中有 R 类资源 m 个，现有 n 个进程互斥使用。若每个进程对 R 资源的最大需求为 w，那么当 m、n、w 分别取表 3-3 中的值时，对于表中的①～⑥种情况，__(1)__ 可能会发生死锁。若将这些情况的 m 分别加上 __(2)__，则系统不会发生死锁。

表 3-3　m、n、w 的取值

	③	②	③	④	⑤	⑥
m	3	3	5	5	6	6
n	2	3	2	3	3	4
w	2	2	3	3	3	2

(1) A. ①②⑤　　B. ③④⑤　　C. ②④⑤　　D. ②④⑥

(2) A. 1、1 和 1　　B. 1、1 和 2　　C. 1、1 和 3　　D. 1、2 和 1

4. 某系统采用请求页式存储管理方案，假设某进程有 6 个页面，系统给该进程分配了 4 个存储块，其页面变换表如表 3-4 所示，表中的状态位等于 1/0 表示页面在内存/不在内存。当该进程访问的页面 2 不在内存时，应该淘汰表中页号为 __(1)__ 的页面。假定页面大小为 4KB，逻辑地址为十六进制 3C18H，该地址经过变换后页帧号为 __(2)__ 。

表 3-4　页面变换表

页　号	页帧号	状态位	访问位	修改位
0	5	1	1	1
1	—	0	0	0
2	—	0	0	0
3	2	1	1	0
4	8	1	1	1
5	12	1	0	0

(1) A. 0　　B. 3　　C. 4　　D. 5

(2) A. 2　　B. 5　　C. 8　　D. 12

5. 假设磁盘每磁道有 18 个扇区，系统刚完成了 10 号柱面的操作，当前移动臂在 13 号柱面上，进程的请求序列如表 3-5 所示。若系统采用 SCAN(扫描)调度算法，则系统响应序列为 __(1)__；若系统采用 CSCAN(单向扫描)调度算法，则系统响应序列为 __(2)__ 。

表 3-5 进程请求序列

请求序列	柱面号	磁头号	扇区号
①	15	8	9
②	20	6	5
③	30	9	6
④	20	10	5
⑤	5	4	5
⑥	2	7	10
⑦	15	8	1
⑧	6	3	4
⑨	8	7	9
⑩	15	10	4

(1) A. ⑦⑩①②④③⑨⑧⑤⑥　　B. ①⑦⑩②③④⑥⑤⑧⑨
　C. ⑦⑩①②④③⑥⑤⑧⑨　　D. ①⑦⑩②③④⑧⑨⑥⑤

(2) A. ⑦⑩①②④③⑨⑧⑤⑥　　B. ①⑦⑩②③④⑥⑤⑧⑨
　C. ⑦⑩①②④③⑥⑤⑧⑨　　D. ①⑦⑩②③④⑧⑨⑥⑤

6. 某磁盘磁头从一个磁道移至另一个磁道需要 10ms。文件在磁盘上非连续存放，逻辑上相邻数据块的平均移动距离为 10 个磁道，每块的旋转延迟时间及传输时间分别为 100ms 和 2ms，则读取一个 100 块的文件需要______ms。

A. 10 200　　B. 11 000　　C. 11 200　　D. 20 200

7. 某文件系统采用多级索引结构，若磁盘块的大小为 512 字节，每个块号需占 3 字节，那么根索引采用一级索引时的文件最大长度为 (1) K 字节；采用二级索引时的文件最大长度为 (2) K 字节。

(1) A. 85　　B. 170　　C. 512　　D. 1024

(2) A. 512　　B. 1024　　C. 14 450　　D. 28 900

8. 若某文件系统的目录结构如图 3-18 所示，假设用户要访问文件 f1.java，且当前工作目录为 Program，则该文件的全文件名为 (1) ，其相对路径为 (2) 。

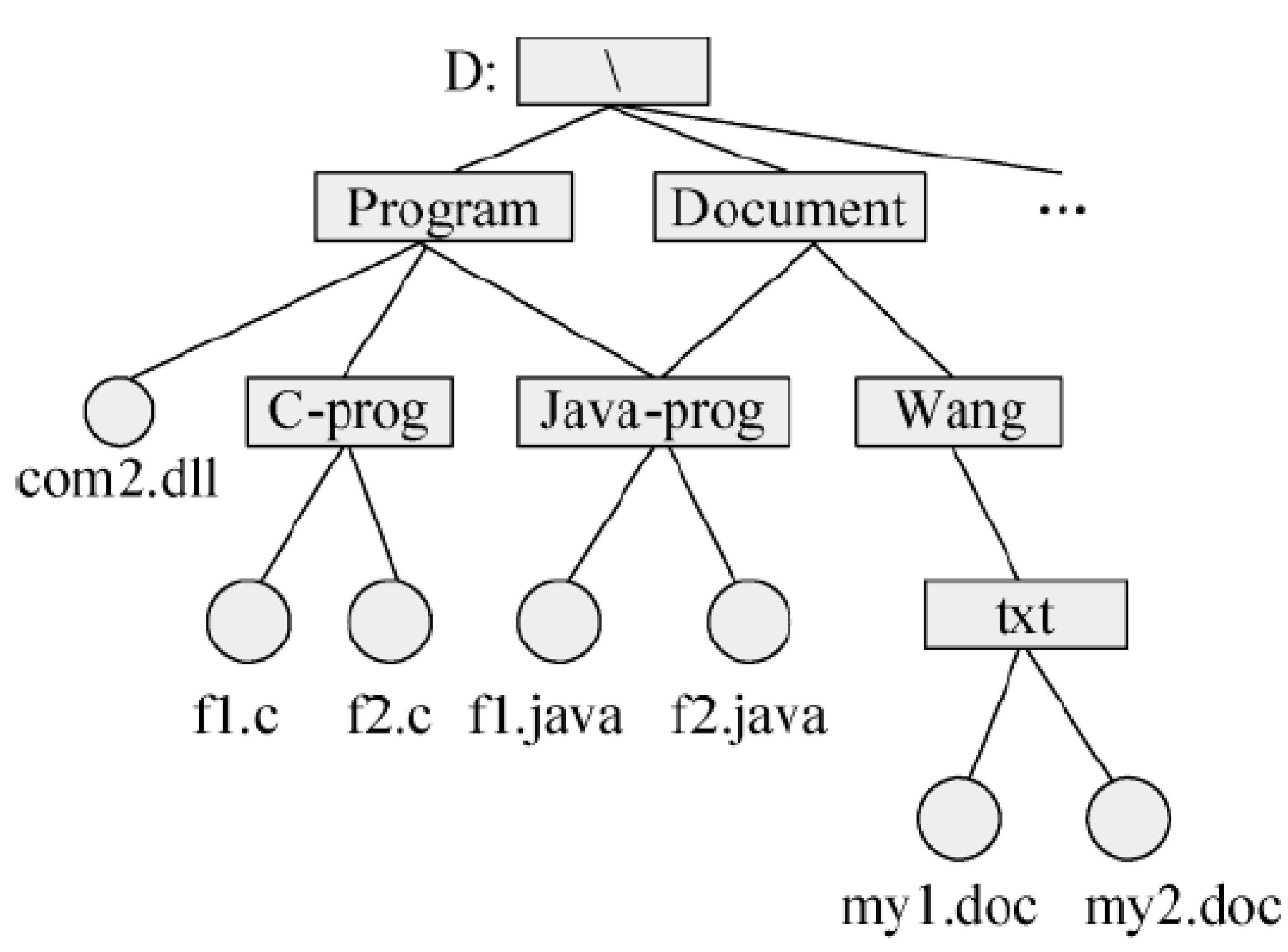


图 3-18 目录结构

(1) A. f1.java　　B. \Document\Java-prog\f1.java

C. D:\Program\Java-prog\f1.java　　D. \Program\Java-prog\f1.java

(2) A. Java-prog\　　B. \Java-prog\

C. Program\Java-prog　　D. \Program\Java-prog\

9. 某文件管理系统在磁盘上建立了位示图(Bitmap)，记录磁盘的使用情况。若系统的字长为 32 位，磁盘上的物理块依次编号为 0，1，2，…，那么 4096 号物理块的使用情况在位示图中的第<u>(1)</u>个字中描述；若磁盘的容量为 200GB，物理块的大小为 1MB，那么位示图的大小为<u>(2)</u>个字。

(1) A. 129　　B. 257　　C. 513　　D. 1025

(2) A. 600　　B. 1200　　C. 3200　　D. 6400

第4章

软件工程基础知识

2014年的《软件设计师教程(第4版)》将“软件开发与运行”拆分为两章：软件工程基础知识和系统开发与运行。

本章要点

- 软件开发生存周期与软件生存周期模型
- 软件开发方法
- 软件开发项目管理
- 软件质量特性(ISO/IEC 9126软件质量模型)
- 软件复杂性的概念及度量方法(McCabe度量法)
- 软件评审(设计质量评审、程序质量评审)
- 软件容错技术
- 软件能力成熟度模型(CMM)
- 统一过程(UP)与极限编程(XP)的基本概念

4.1　软件工程概述

考核说明：本节主要介绍软件危机、软件工程、软件生存周期、软件过程，软件过程中的软件能力成熟度模型(CMM)、统一过程(UP)是考核重点，需要掌握。

4.1.1　软件危机

软件危机主要表现在以下几个方面。

(1) 软件开发费用和进度失控。

(2) 软件的可靠性差。

(3) 生产出来的软件难以维护。

(4) 用户对“已完成”的系统不满意现象经常发生。

(5) 软件产品质量难以保证。

(6) 软件文档不完备，并且存在文档内容与软件产品不符的情况。

随着计算机应用需求的不断增长，软件的规模也越来越大，软件危机不仅没有消失，还有加剧之势。主要表现在：

① 软件成本在计算机系统总成本中所占的比例居高不下，且逐年上升。

② 软件开发生产率提高的速度远远跟不上计算机应用迅速普及深入的需要。

1968 年，在德国召开的 NATO(North Atlantic Treaty Organization，北大西洋公约组织)会议上首次提出了“软件工程”这个名词，希望用工程化的原则和方法来克服软件危机。

软件工程是指应用计算机科学、数学及管理科学等原理，以工程化的原则和方法来解决软件问题的工程，其目的是提高软件生产率、提高软件质量、降低软件成本。软件工程涉及软件开发、维护、管理等多方面的原理、方法、工具与环境。

4.1.2　计算机软件

软件是由计算机程序和程序设计的概念发展演化而来的，是在程序和程序设计发展到一定规模并且逐步商品化的过程中形成的。计算机软件是指计算机系统中的程序及其文档。按照应用软件的应用领域，可将计算机软件分为七大类：

(1) 系统软件；

(2) 应用软件；

(3) 工程/科学软件；

(4) 嵌入式软件；

(5) 产品线软件；

(6) Web 应用软件；

(7) 人工智能软件。

4.1.3 软件工程基本原理

著名软件工程专家 B.Boehm 综合有关专家和学者的意见并总结了多年来开发软件的经验，于 1983 年在一篇论文中提出了软件工程的 7 条基本原理。

1. 用分阶段的生命周期计划严格管理

Boehm 认为，在整个软件生命周期中应指定并严格执行 6 类计划：项目概要计划、里程碑计划、项目控制计划、产品控制计划、验证计划、运行维护计划。

2. 坚持进行阶段评审

软件的质量保证工作不能等到编码结束之后再进行，应坚持进行严格的阶段评审，以便尽早发现错误。

3. 实行严格的产品控制

实践中，需求的改动往往是不可避免的，这就要求我们要采用科学的产品控制技术来顺应这种要求，也就是要采用变动控制，又叫基准配置管理。当需求变动时，其他各个阶段的文档或代码随之相应变动，以保证软件的一致性。

4. 采纳现代程序设计技术

采用先进的技术既可以提高软件开发的效率，又可以减少软件维护的成本。

5. 结果应能清楚地审查

为更好地进行管理，应根据软件开发的总目标及完成期限，尽量明确地规定开发小组的责任和产品标准，从而使所得到的标准能清楚地审查。

6. 开发小组的人员应少而精

这一条基于两点原因：高素质开发人员的效率比低素质开发人员的效率要高几倍到几十倍，开发工作中犯的错误也要少得多；当开发小组为 N 人时，可能的通信信道为 $N(N-1)/2$，可见随着人数 N 的增大，通讯开销将急剧增大。

7. 承认不断改进软件工程实践的必要性

根据这条原理，不仅要积极采纳新的软件开发技术，还要注意不断总结经验，收集进度和消耗等数据，进行出错类型和问题报告统计。这些数据既可以用来评估新的软件技术的效果，也可以用来指明必须着重注意的问题和应该优先进行研究的工具和技术。

4.1.4 软件生存周期

同任何事物一样，软件也有一个孕育、诞生、成长、成熟、衰亡的生存过程，我们称其为计算机软件的生存周期。通常，软件生存周期包括以下几个阶段。

(1) 可行性分析与项目开发计划：该阶段主要确定软件的开发目标及其可行性；参加

人员有用户、项目负责人和系统分析师；产生的主要文档有可行性分析报告和项目开发计划。

(2) 需求分析：该阶段的任务是确定软件系统的功能、性能、数据和界面等要求，从而确定系统的逻辑模型；参加人员有用户、项目负责人和系统分析师；产生的主要文档有软件需求说明书。

(3) 概要设计：该阶段要把确定的各项功能需求转换成需要的体系结构，还要设计该项目的应用系统的总体数据结构和数据库结构；参加人员有系统分析师和软件设计师；产生的主要文档有概要设计说明书。

(4) 详细设计：主要任务是对每个模块完成的功能进行具体描述，要把功能描述转变为精确的、结构化的过程描述；参加人员有软件设计师和程序员；产生的主要文档有详细设计文档。

(5) 编码：把每个模块的控制结构转换成计算机可接受的程序代码。

(6) 测试：在设计测试用例的基础上检查软件的各个组成部分；参加人员通常是另一部门(或单位)的软件设计师或系统分析师；产生的主要文档有软件测试计划、测试用例和软件测试报告。

(7) 维护：已交付的软件正式投入使用后，便进入软件维护阶段，它可以持续几年甚至几十年。

4.1.5 软件过程

软件过程能力评估是软件过程改进和降低软件风险的需要。

1. 软件能力成熟度模型(CMM)简介

CMM 将软件组织的过程能力分成 5 个成熟度级别：初始级、可重复级、已定义级、已管理级和优化级。由低到高，软件开发生产精度越来越高，每单位工程的生产周期越来越短。

(1) 初始级。软件过程是无序的，有时甚至是混乱的，对过程几乎没有定义，成功取决于个人努力。

(2) 可重复级。建立了基本的项目管理过程来跟踪费用、进度和功能特性；制定了必要的过程纪律，能重复早先类似应用项目取得的成功。

(3) 已定义级。已将软件管理和工程两方面的过程文档化、标准化，并综合成该组织的标准软件过程。所有项目均使用经批准、剪裁的标准软件过程来开发和维护软件。

(4) 已管理级。收集对软件过程和产品质量的详细度量，对软件过程和产品都有定量的理解和控制。

(5) 优化级。过程的量化反馈和先进的新思想、新技术促使过程不断改进。

2. 能力成熟度模型(CMMI)

CMMI 是若干过程模型的综合和改进，是支持多个工程学科和领域的、系统的、一致的过程改进框架，能适应现代工程的特点和需要，能提高过程的质量和工作效率。

CMMI 提供了两种表示方法：阶段式模型和连续式模型。

阶段式模型的结构类似于 CMM，它关注组织的成熟度。连续式模型关注每个过程域的能力，一个组织对不同的过程域可以达到不同的过程域能力等级(Capability Level，CL)。CMMI 中包括 6 个过程域能力等级，等级号为 0～5。

3. 统一过程(UP)

统一过程(UP)模型是一种“用例和风险驱动，以架构为中心，迭代并且增量”的开发过程，由 UML 方法和工具支持。

统一过程包括 4 个阶段：初始阶段、精化阶段、构建阶段、移交阶段。4 个技术阶段由主要里程碑所终止。

统一过程的典型代表是 RUP(Rational Unified Process，Rational 统一过程)。RUP 是 UP 的商业扩展，完全兼容 UP，但比 UP 更完整、更详细。

真题链接

【例 4-1】以下关于 CMM 的叙述中，不正确的是__(30)__。(2014 年 11 月真题 30)

(30) A. CMM 是指软件过程能力成熟度模型

B. CMM 根据软件过程的不同成熟度划分了 5 个等级，其中，1 级被认为成熟度最高，5 级被认为成熟度最低

C. CMMI 的任务是将已有的几个 CMM 模型结合在一起，使之构造成为“集成模型”

D. 采用更成熟的 CMM 模型，一般来说可以提高最终产品的质量

【解析】CMM 模型指“能力成熟度模型”，其英文全称为 Capability Maturity Model for Software，英文缩写为 SW-CMM，简称 CMM。它是对于软件组织在定义、实施、度量、控制和改善其软件过程的实践中各个发展阶段的描述。CMM 的核心是把软件开发视为一个过程，并根据这一原则对软件开发和维护进行过程监控和研究，以使其更加科学化、标准化、使企业能够更好地实现商业目标。CMM 模型分为 5 个等级(成熟度由低到高)：①初始级，②可重复级，③已定义级，④定量管理级，⑤优化级。1 级成熟度最低，5 级最高，故选项 B 错误。

【答案】B

【例 4-2】以下关于统一过程 UP 的叙述中，不正确的是__(29)__。(2014 年 5 月真题 29)

(29) A. UP 是以用例和风险为驱动，以架构为中心，迭代并且增量的开发过程

B. UP 定义了四个阶段，即起始、精化、构建和确认阶段

C. 每次迭代都包含计划、分析、设计、构造、集成、测试以及内部和外部发布

D. 每个迭代有五个核心工作流

【解析】统一过程(UP)的基本特征是用例和风险驱动，以架构为中心，受控的迭代式的增量开发。UP 定义了四个阶段。

起始阶段(inception)：该阶段的主要目的是建立项目的范围和版本，确定项目目标的可行性和稳定性，提交结果包括需求和用例。

精化阶段(elaboration)：该阶段的目的是对问题域进行分析，建立系统需求和架构，确定实现的可行性和稳定性，提交结果包括系统架构，问题领域、修改后的需求及项目开发计划等相关文档。

构建阶段(construction)：增量式开发可以交付给用户的软件产品。

移交阶段(transition)：目的是将软件产品交付给用户。

【答案】B

4.2　软件过程模型

考核说明：本节主要介绍瀑布模型、增量模型、演化模型、喷泉模型、基于构件的开发模型和形式化方法模型。其中，瀑布模型、增量模型是考核重点，需要掌握。

4.2.1　瀑布模型

该模型给出了软件生存周期各阶段的固定顺序，上一阶段完成后才能进入下一阶段，整个过程就像流水下泻，故称为瀑布模型，如图 4-1 所示。这种模型缺乏灵活性，无法通过开发活动来澄清本来不够明确的需求。

瀑布模型的一个变体是 V 模型，如图 4-2 所示。V 模型描述了质量保证活动和沟通、建模相关活动以及早期构建相关的活动之间的关系，提供了一种将验证确认活动应用于早期软件工程工作中的方法。

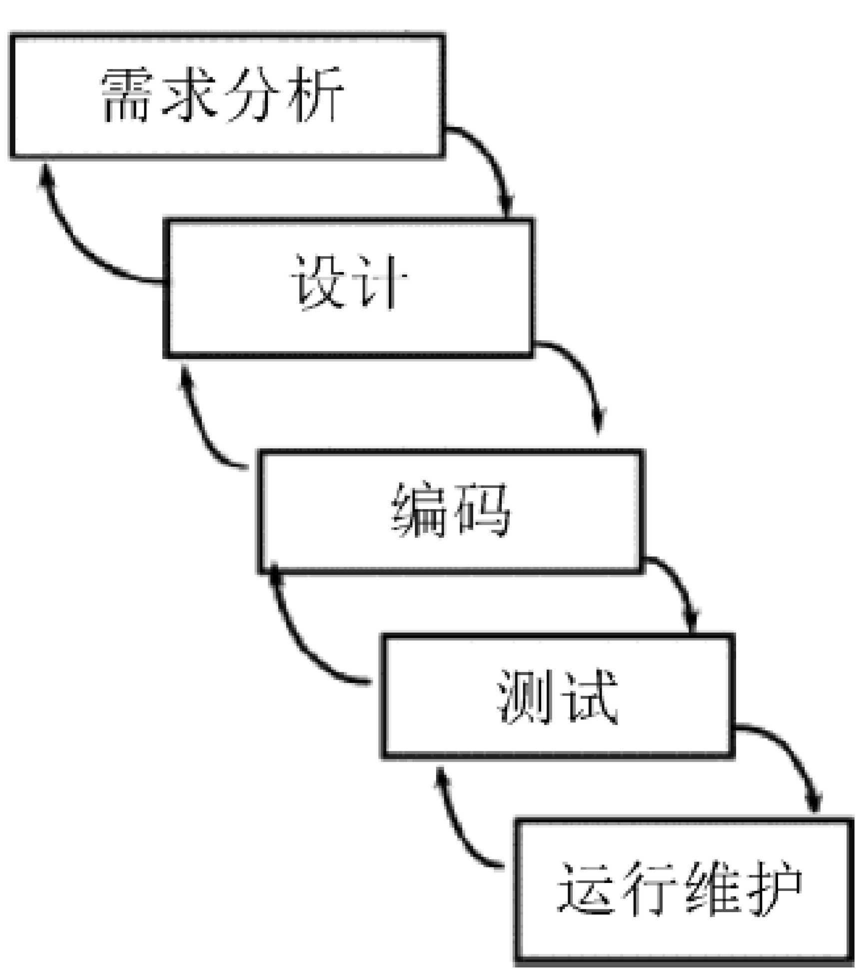


图 4-1　瀑布模型

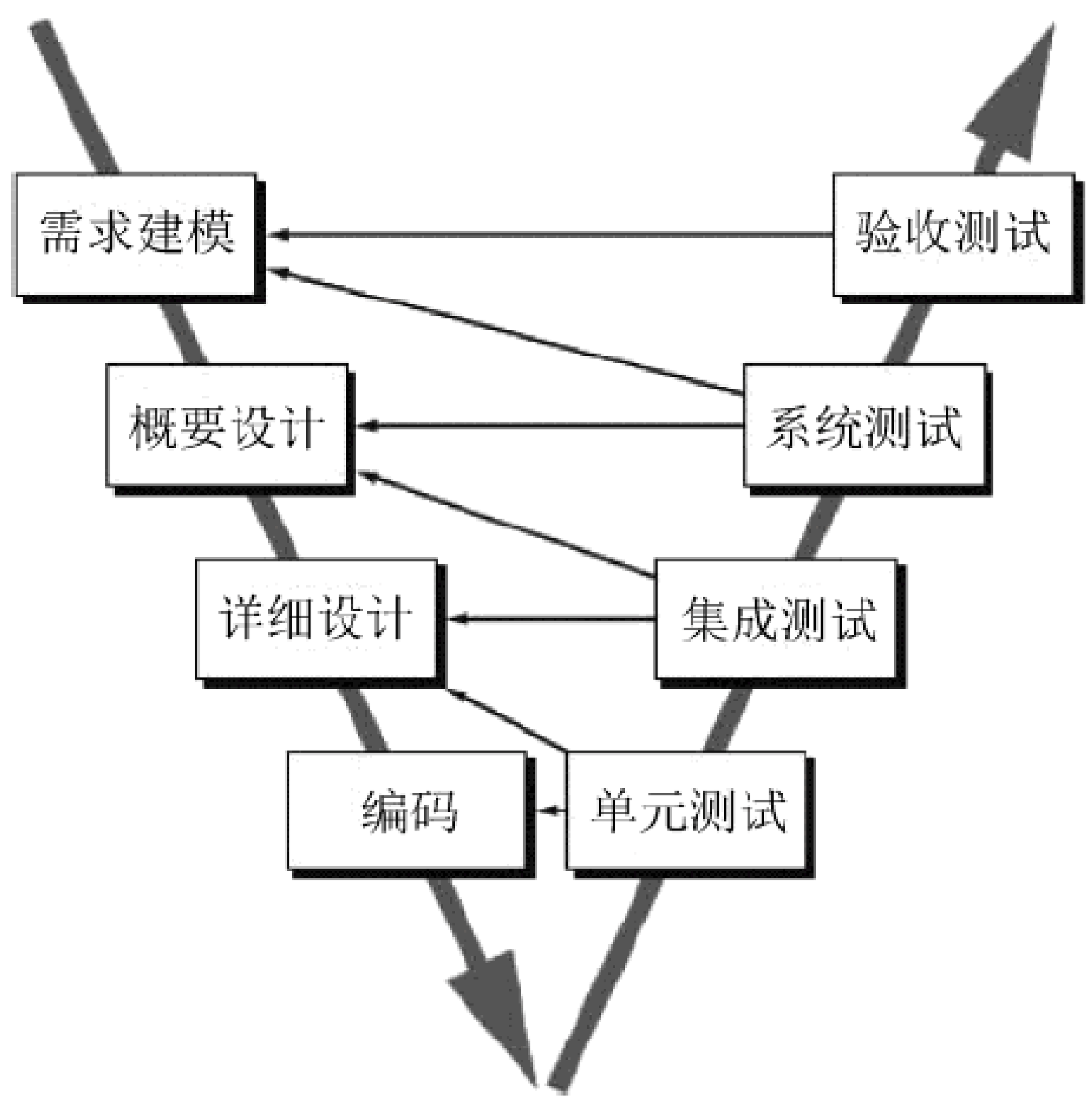


图 4-2　V 模型

真题链接

【例 4-3】某公司要开发一个软件产品，产品的某些需求是明确的，而某些需求则需要进一步细化。由于市场竞争的压力，产品需要尽快上市，则开发该软件产品最不适合采用<u>(30)</u>模型。(2014 年 5 月真题 30)

(30) A. 瀑布　　B. 原型　　C. 增量　　D. 螺旋

【解析】瀑布模型将软件生命周期划分为需求分析、软件设计、程序编写、软件测试和运行维护等基本活动，并且规定了它们自上而下、相互衔接的固定次序，如同瀑布流水，逐级下落。瀑布模型要求待开发的系统需求是完整的、简明的、一致的，其突出的缺点是不适应用户需求的变化，开发模型是线性的，

用户只有等到整个过程的末期才能见到开发成果。对于需求不确定的软件产品不适合采用瀑布模型。

【答案】A

4.2.2　增量模型

增量模型融合了瀑布模型的基本成分(重复应用)和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”，如图 4-3 所示。当使用增量模型时，第 1 个增量往往是核心的产品，即第 1 个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。

增量模型的特点是引进了增量包的概念，无须等到所有需求都出来，只要某个需求的增量包出来即可进行开发。虽然某个增量包可能还需要进一步适应客户的需求并且更改，但只要这个增量包足够小，其影响对整个项目来说是可以承受的。

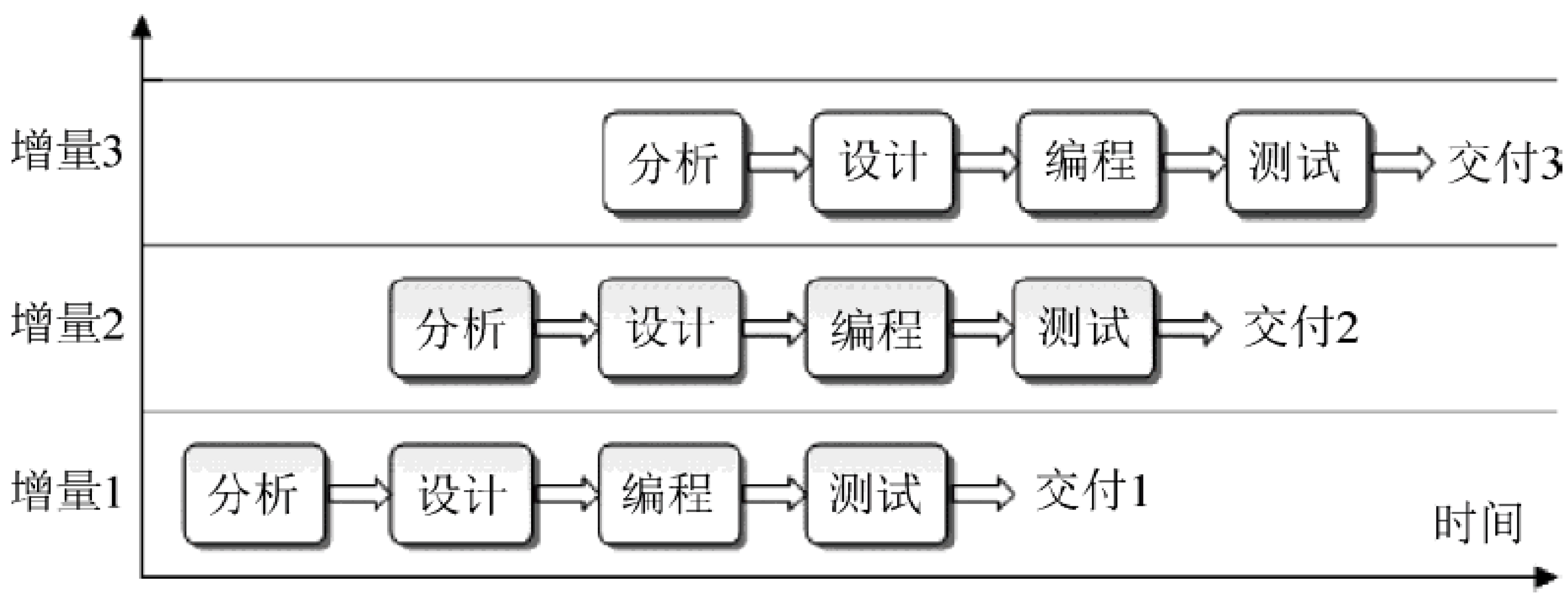


图 4-3　增量模型

真题链接

【例 4-4】以下关于增量模型的叙述中，正确的是<u>(29)</u>。(2014 年 11 月真题 29)

(29) A. 需求被清晰定义　　B. 可以快速构造核心产品

C. 每个增量必须要进行风险评估　　D. 不适宜商业产品的开发

【解析】增量模型融合了瀑布模型的基本成分(重复应用)和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时，第1个增量往往是核心的产品，即第1个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。利用增量模型，能够在很短的时间内向用户提交一些有用的工作产品，可以快速构造核心产品。增量模型能够较好地应对需求变化。若需求被清晰定义，则应采用瀑布模型。

【答案】B

4.2.3　演化模型

演化模型是迭代的过程模型，使得软件开发人员能够逐步开发出更完整的软件版本。

演化模型特别适用于对软件需求缺乏准确认识的情况。典型的演化模型有原型模型和螺旋模型等。

1. 原型模型

在获取一组基本的需求后，通过快速分析构造出该软件的一个初始可运行版本，这个初始的软件通常称为原型，然后根据用户在使用原型的过程中提出的意见和建议对原型进行改进，获得原型的新版本。重复这一过程，最终可得到令用户满意的软件产品，如图 4-4 所示。该模型适用于对软件需求缺乏准确认识的情况。

2. 螺旋模型

将瀑布模型和演化模型相结合就成了螺旋模型。这种模型综合了瀑布模型和演化模型的优点，并增加了风险分析。螺旋模型包括四个方面的活动：制订计划、风险分析、实施工程、客户评估，如图 4-5 所示。螺旋模型强调风险分析，使得开发人员和用户对每个演化层出现的风险有所了解，从而做出应有的反应。因此，该模型特别适用于庞大、复杂并且具有高风险的系统。

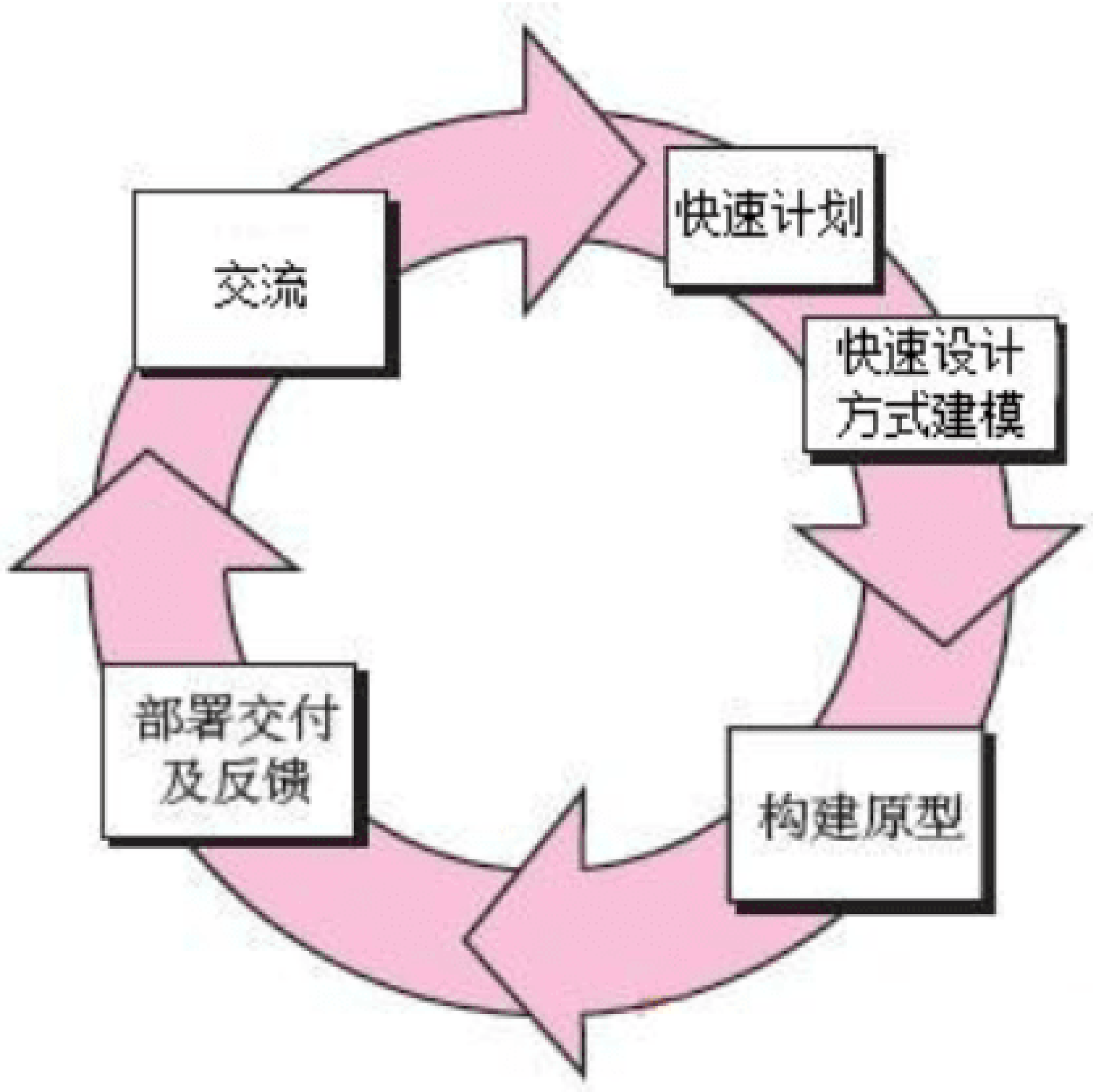


图 4-4　原型模型

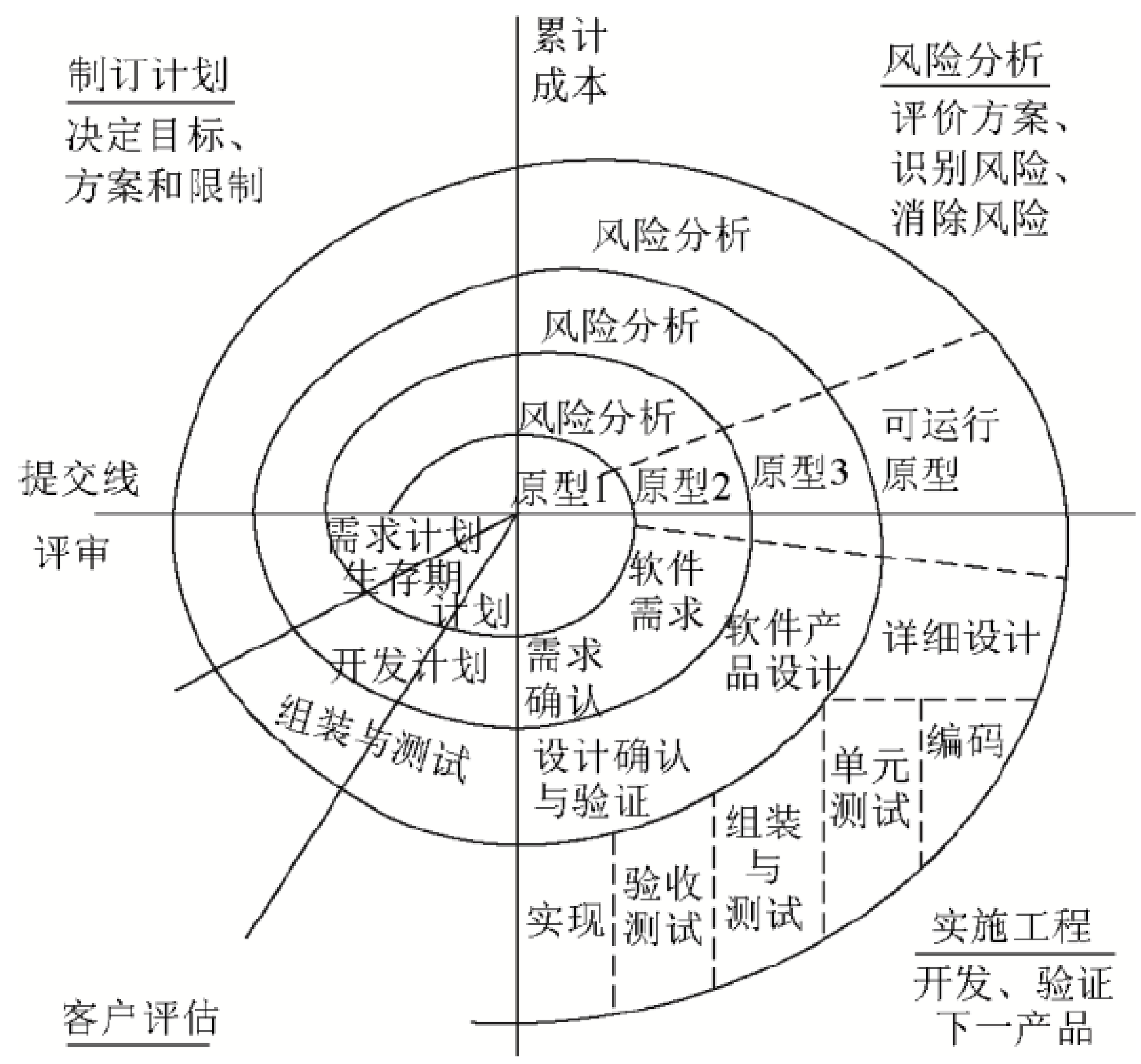


图 4-5　螺旋模型

4.2.4　喷泉模型

喷泉模型主要用于描述面向对象的开发过程。该模型具有迭代和无间隙特性，如图 4-6 所示。迭代意味着模型中的开发活动常常需要重复多次，在迭代中不断完善软件系统。无间隙是指在开发活动之间不存在明显的边界，允许开发活动交叉、迭代地进行。由于喷泉模型在各个开发阶段是重叠的，在开发过程中需要大量的开发人员，不利于项目的管理。此外，这种模型要求严格管理文档，使得审核的难度加大。

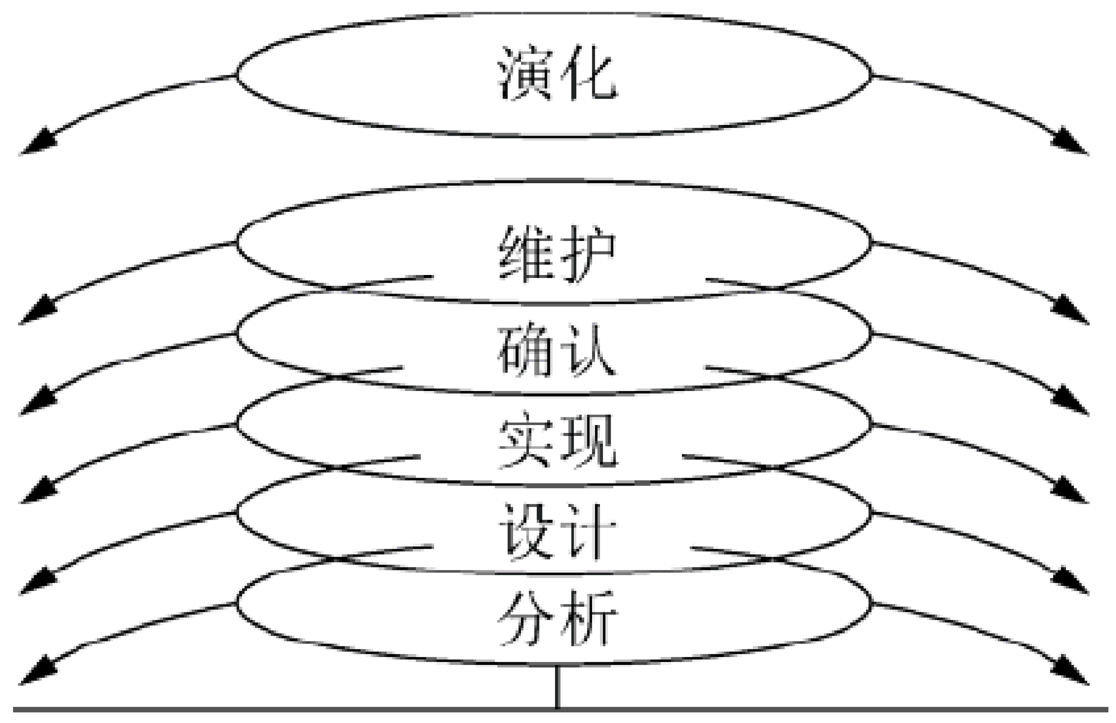


图 4-6　喷泉模型

4.2.5　基于构件的开发模型

基于构件的开发模型具有许多螺旋模型的特点，它本质上是演化模型，需要以迭代方式构建软件。其不同之处在于，基于构件的开发模型采用预先打包的软件构件开发应用系统。一种基于构件的开发模型，包括领域工程和应用系统工程两部分。

领域工程是在构造一个特定领域内的系统或者系统的某些部分时，以可重用方面的形式(也就是说，可重用的工作产物)，收集、组织并保存过去的经验的活动，以及在构造新系统时，提供一种充分的方法来重用这些资源(也就是说，获取、限定、改造、装配等等)。领域工程的目的是构建领域模型、领域基准体系结构和可复用构件库。

应用系统工程的目的是使用可复用构件组装应用系统。

4.2.6　形式化方法模型

形式化方法模型(formal methods model)的主要活动是生成计算机软件形式化的数学规格说明。形式化方法使软件开发人员可以应用严格的数学符号来说明、开发和验证基于计算机的系统。

形式化方法提供了一种机制，使得在软件开发中可以避免一些问题，而这些问题在使用其他软件工程模型时难以解决。使用形式化方法，诸如歧义性问题、不完整问题、不一致问题都能够更容易地被发现和改正——不是依靠特定的评审，而是应用数学分析的方法。在设计阶段，形式化方法是程序验证的基础，使软件开发人员能够发现和改正一些往往被忽略的问题。

这种方法的一个变型是净室软件工程(cleanroom software engineering)，这一软件工程方法目前已应用于一些软件开发机构。

4.3　软件开发方法

考核说明： 本节主要介绍结构化方法、Jackson 方法、原型方法等各种软件开发方法，其中敏捷方法是重点，需要掌握。

4.3.1 结构化方法

结构化方法是由 E.Yourdon 和 L.L.Constantine 提出的，即所谓的 SASD 方法，也可称为面向功能的软件开发方法或面向数据流的软件开发方法。

结构化方法首先用结构化分析(SA)对软件进行需求分析，然后用结构化设计(SD)方法进行总体设计，最后是结构化编程(SP)。它给出了两类典型的软件结构(变换型和事务型)使软件开发的成功率大大提高。

结构化方法总的指导思想是自顶向下、逐层分解，它的基本原则是功能的分解与抽象。它特别适合于解决数据处理领域的问题，但是不适合解决大规模的、特别复杂的项目，且难以适应需求的变化。

4.3.2 Jackson 方法

Jackson 方法是最典型的面向数据结构的软件开发方法，Jackson 方法把问题分解为可由三种基本结构形式表示的各部分的层次结构。三种基本的结构形式就是顺序、选择和重复。这三种结构可以进行组合，形成复杂的结构体系。这一方法从目标系统的输入、输出数据结构入手，导出程序框架结构，再补充其他细节，就可得到完整的程序结构图。这一方法对输入、输出数据结构明确的中小型系统特别有效，如商业应用中的文件表格处理。该方法也可与其他方法结合，用于模块的详细设计。

4.3.3 原型方法

产生原型化方法的原因很多，主要是因为随着我们系统开发经验的增多，发现并非所有的需求都能够预先定义而且反复修改是不可避免的。当然能够采用原型化方法是因为开发工具的快速发展，比如用 VB、Delphi 等工具我们可以迅速开发出一个能够让用户看得见、摸得着的系统框架，这样，对于计算机不是很熟悉的用户就可以根据这个样板提出自己的需求。

原型方法比较适合于用户需求不清、需求经常变化的情况。当系统规模不是很大也不太复杂时，采用该方法比较好。

4.3.4 面向对象开发方法

面向对象方法的基本出发点是尽可能地按照人类认识世界的方法和思维方式来分析和解决问题。面向对象方法包括面向对象分析、面向对象设计和面向对象实现。面向对象方法有 Booch 方法、Coad 方法和 OMT 方法等。

Booch 最先描述了面向对象的软件开发方法的基础问题，指出面向对象开发是一种根本不同于传统的功能分解的设计方法。面向对象的软件分解更接近人对客观事务的理解，而功能分解只通过问题空间的转换来获得。

Coad 方法是 1989 年 Coad 和 Yourdon 提出的面向对象开发方法。该方法的主要优点是

通过多年来大系统开发的经验与面向对象概念的有机结合，在对象、结构、属性和操作的认定方面，提出了一套系统的原则。该方法完成了从需求角度进一步进行类和类层次结构的认定。尽管 Coad 方法没有引入类和类层次结构的术语，但事实上已经在分类结构、属性、操作、消息关联等概念中体现了类和类层次结构的特征。

OMT 方法是 1991 年由 James Rumbaugh 等 5 人提出来的。开发工作的基础是对真实世界的对象建模，然后围绕这些对象使用分析模型来进行独立于语言的设计，面向对象的建模和设计促进了对需求的理解，有利于开发出更清晰、更容易维护的软件系统。

4.3.5　敏捷方法

敏捷开发的总体目标是通过“尽可能早地、持续地对有价值的软件的交付”使客户满意。

(1) 极限编程(XP)。XP 是一种轻量级(敏捷)、高效、低风险、柔性、可预测的、科学的软件开发方式。它由价值观、原则、实践和行为四个部分组成，彼此相互依赖、关联，并通过行为贯穿于整个生存周期。

4 个价值观：①沟通；②简单性；③反馈；④勇气。

5 个原则：①快速反馈；②简单性假设；③逐步修改；④提倡更改；⑤优质工作。

12 个最佳实践：①计划游戏(快速制订计划，随着细节的不断变化而完善)；②小型发布(系统的设计要能够尽可能早地交付)；③隐喻(找到合适的比喻传达信息)；④简单设计(只处理当前的需求，使设计保持简单)；⑤测试先行(先写测试代码，然后再编写程序)；⑥重构(重新审视需求和设计，重新明确地描述它们以符合新的和现有的需求)；⑦结队编程；⑧集体代码所有制；⑨持续集成(可以按日甚至按小时为客户提供可运行的版本)；⑩每周工作 40 个小时；⑪现场客户；⑫编码标准。

(2) 水晶法(Crystal)。水晶法是 Alistair Cockburn 在 20 世纪 90 年代末提出的，他相信不同类型的项目需要不同的方法。

(3) 并列争球法(Scrum)。Scrum 是一种迭代的增量化过程，用于产品开发或工作管理。它是一种可以集合各种开发实践的经验化过程框架。多个自组织和自治的小组并行地递增实现产品。协调是通过简短的日常情况会议来进行的，就像橄榄球中的“并列争球”。

(4) 自适应软件开发(ASD)。ASD(Adaptive Software Development，自适应软件开发)由 Jim Highsmith 在 1999 年正式提出。ASD 强调开发方法的适应性(Adaptive)，这一思想来源于复杂系统的混沌理论。ASD 不像其他方法那样有很多具体的实践做法，它更侧重为 ASD 的重要性提供最根本的基础，并从更高的组织和管理层次来阐述开发方法为什么要具备适应性。

4.4　软件工具和软件开发环境

考核说明：本节主要介绍软件工具和软件开发环境，不是考核的重点，在最近几次考试中没有出现相关考题，了解即可。

4.4.1 软件工具

软件工具可以分为如下几类。

(1) 软件开发工具。包括需求分析工具、设计工具和编码与排错工具。

(2) 软件维护工具。包括版本控制工具、文档分析工具、开发信息库工具、逆向工程工具和再工程工具。

(3) 软件管理和支持工具。包括项目管理工具、配置管理工具和软件评价工具。

4.4.2 软件开发环境

软件开发环境是支持软件产品开发的软件系统。它由软件工具集和环境集成机制构成，前者用来支持软件开发的相关过程、活动和任务；后者为工具集成和软件开发、维护和管理提供统一的支持，它通常包括数据集成、控制集成和界面集成。

软件开发环境的特征如下。

(l) 环境的服务是集成的。

(2) 环境应支持小组工作方式，并为其提供配置管理。

(3) 环境的服务可用于支持各种软件开发活动，包括分析、设计、编程、测试、调试和文档等。

4.5 软件项目管理

考核说明：本节是考核的重点，Gantt 图、PERT 图、软件质量管理是重中之重，需要掌握好。

4.5.1 软件项目管理的范围

有效的软件项目管理集中在 4 个 P 上，即人员(Person)、产品(Product)、过程(Procedure)和项目(Project)。

1. 人员

人员是软件工程项目的基本要素和关键因素，一般来说，包括项目管理人员、高级管理人员、开发人员、客户和最终用户。其中，项目管理人员负责软件项目的管理工作，其负责人通常称为项目经理。高级管理人员负责提出项目的目标并对业务问题进行定义。

2. 产品

在进行项目计划之前，应该首先定义项目范围，其中包括建立产品的目的和范围、可选的解决方案、技术或管理的约束等。

3. 过程

软件项目强调的是对其进行过程控制，通常将项目分解为任务、子任务等，其分解准则是基于软件工程的过程。按照软件过程的要求，一个项目团队要选择一个适合于待开发软件的过程模型，项目团队必须决定制定一个完整的计划来反映框架活动中所需完成的工作任务。

4. 项目

Reel 提出了包含如下 5 个部分常识的软件项目方法。

(1) 明确目标及过程。

(2) 保持动力。

(3) 跟踪进展。

(4) 做出明智的决策。

(5) 进行事后分析。

4.5.2 软件项目估算

软件项目常用的估算方法有下列 3 种：

(1) 基于已经完成的类似项目进行估算。

(2) 基于分解技术进行估算。分解技术包括问题分解和过程分解。

(3) 基于经验估算模型的估算。典型的经验估算模型有 IBM 估算模型、COCOMO 模型和 Putnam 模型。

下面介绍成本估算方法和几种成本估算模型。

1. 成本估算方法

常见的成本估算方法有自顶向下估算方法、自底向上估算方法、差别估算方法，以及专家估算法、类推估算法、算式估算法。

2. 成本估算模型

Putnam 模型是一种动态多变量模型，它假设在软件开发的整个生存期中工作量有特定的分布。

结构性成本模型 COCOMO 是最精确、最易于使用的成本估算模型之一。该模型可以分为如下几种。

- 基本 COCOMO 模型。它是一个静态单变量模型，它是对整个软件系统进行估算。
- 中级 COCOMO 模型。它是一个静态多变量模型，它将软件系统模型分为系统和部件两个层次，系统由部件构成，它把软件开发所需人力(成本)看作程序大小和一系列“成本驱动属性”的函数。
- 详细 COCOMO 模型。它将软件系统模型分为系统、子系统和模块三个层次，它除包括中级模型所考虑的因素外，还要考虑需求分析、软件设计等每一步的成本驱动属性的影响。

COCOMO 模型已经演化成更全面的估算模型，称为 COCOMOII。COCOMOII 也是一种层次结构的估算模型，被分为 3 个阶段性模型。

- 应用组装模型。在软件工程的前期阶段使用。
- 早期设计阶段模型。在需求已经稳定并且基本的软件体系结构已经建立时使用。
- 体系结构阶段模型。在软件的构造过程中使用。

真题链接

【例 4-5】_(17)_ 软件成本估算模型是一种静态单变量模型，用于对整个软件系统进行估算。(2014 年 5 月真题 17)

(17) A. Putnam　　B. 基本 COCOMO　　C. 中级 COCOMO　　D. 详细 COCOMO

【解析】COCOMO 用 3 个不同层次的模型来反映不同程度的复杂性，它们分别为：

基本模型 (Basic Model)：是一个静态单变量模型，它用一个以项目的源代码行估计值(LOC)为自变量的函数来计算软件开发工作量。

中级模型 (Intermediate Model)：在用以 LOC 为自变量的函数计算软件开发工作量的基础上，再用涉及产品、硬件、人员、项目等方面属性的影响因素来调整工作量的估算。

详细模型 (Detailed Model)：包括中级 COCOMO 模型的所有特性，但用上述各种影响因素调整工作量估算时，还要考虑对软件工程过程中分析、设计等各步骤的影响。

【答案】B

4.5.3 进度管理

软件开发项目的进度安排有两种方式：系统最终交付日期已经确定，软件开发部门必须在规定期限内完成；系统最终交付日期只确定了大致的年限，最后交付日期由软件开发部门确定。

1. 进度管理的基本原则

(1) 划分。项目必须被划分成若干个可以管理的活动和任务。

(2) 相互依赖性。划分后的各个活动或任务之间的相互依赖关系必须是明确的。

(3) 时间分配。必须为每个被调度的任务分配一定数量的工作单位，必须为每个任务制定开始和结束日期。

(4) 工作量确认。项目管理者必须确保在任意时段中分配的人员数量不会超过项目团队中的总人数。

(5) 确定责任。

(6) 明确输出结果。安排了进度计划的每个任务都应该有一个明确的输出结果。

(7) 确定里程碑。当一个或多个工作产品经过质量评审并且得到认可时，标志着一个里程碑的完成。

2. 进度安排

进度安排的常用图形描述方法有甘特(Gantt)图和计划评审技术(PERT)图。

1)　Gantt 图

Gantt 图中横坐标表示时间，纵坐标表示任务，图中的水平线段则表示对一个任务的进度安排，线段的起点和终点对应在横坐标上的时间分别表示该任务的开始时间和结束时间，线段的长度表示完成该任务所需的时间。

Gantt 图能清晰地描述每个任务从何时开始、到何时结束以及各个任务之间的并行性，但是它不能清晰地反映出各任务之间的依赖关系，难以确定整个项目的关键所在，也不能反映计划中有潜力的部分。

2)　PERT 图

PERT 图是一个有向图，箭头表示任务，它可以表示完成该任务所需的时间；箭头指向结点表示流入结点的任务的结束，并开始流出结点的任务，这里把结点当成事件。只有当流入该结点的所有任务都结束时，结点所表示的事件才出现，流出结点的任务才可以开始。事件本身不消耗时间和资源，它仅表示某个时间点。一个事件有一个事件号和出现该事件的最早时刻和最迟时刻。每个任务还有一个松弛时间，表示在不影响整个工期的前提下，完成该任务有多少机动余地。

PERT 图不仅给出了每个任务的开始时间、结束时间和完成该任务所需的时间，还给出了任务之间的关系，即哪些任务完成后才能开始另外一些任务，以及如期完成整个工程的关键路径。松弛时间则反映了完成某些任务时可以推迟其开始时间或延长其所需的完成时间。但是 PERT 图不能反映任务之间的并行关系。

真题链接

【例 4-6】图 4-7 是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示活动，边的权重表示活动的持续时间，则里程碑__(19)__在关键路径上。活动 GH 的松弛时间是__(20)__。(2014 年 11 月真题 19～20)

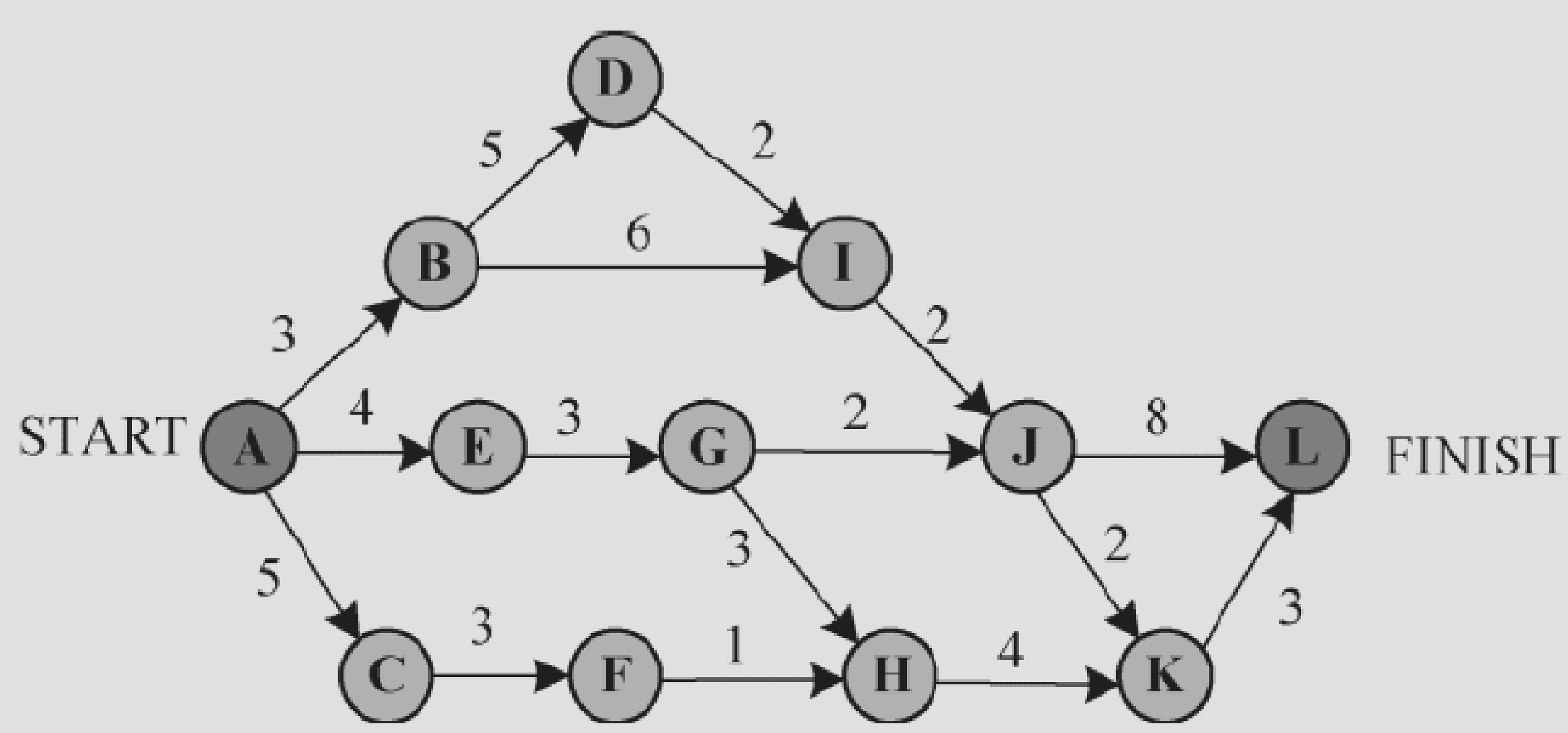


图 4-7　项目的活动图

(19) A. B　　B. E　　C. C　　D. K

(20) A. 0　　B. 1　　C. 2　　D. 3

【解析】关键路径为 A-B-D-I-J-L。GH 活动最早开始时间是 7，最晚开始时间是 10，所以松弛时间是 3。

【答案】(19)A；(20)D

【例 4-7】以下关于进度管理工具 Gantt 图的叙述中，不正确的是__(18)__。(2014 年 5 月真题 18)

(18) A. 能清晰地表达每个任务的开始时间、结束时间和持续时间

B. 能清晰地表达任务之间的并行关系

C. 不能清晰地确定任务之间的依赖关系

D. 能清晰地确定影响进度的关键任务

【解析】甘特图内在思想简单，即以图示的方式通过活动列表和时间刻度形象地表示出任何特定项目的活动顺序与持续时间。基本是一个线条图，横轴表示时间，纵轴表示活动(项目)，线条表示在整个期间中计划和实际的活动完成情况。它可以直观地表明任务计划在什么时候进行，以及实际进展与计划要求的对比。管理者由此可便捷地弄清一项任务(项目)还剩下哪些工作要做，并可评估工作进度。

优点：能清晰地描述每个任务从何时开始，到何时结束以及各个任务之间的并行性。

缺点：不能清晰地反映出个任务之间的依赖关系，难以确定整个项目的关键所在，也不能反映计划中有潜力的部分。

【答案】D

4.5.4 软件项目的组织

组织原则有以下 3 条：

(1) 尽早落实责任。

(2) 减少交流接口。

(3) 责权均衡。

可以按软件项目对软件人员分组，如需求分析组、设计组、编码组、测试组、维护组等，为了控制软件的质量，还可以有质量保证组。

程序设计小组的组织形式可以有多种，如主程序员制小组、民主制小组、层次式小组等。

主程序员制小组由一名主程序员、若干名程序员、一名后援工程师和一名资料员组成。主程序员通常由高级工程师担任，负责小组的全部技术活动，进行任务的分配，协调技术问题，组织评审，必要时也设计和实现项目中的关键部分。主程序员制小组突出了主程序员的领导作用，小组内的通信主要体现在主程序员与程序员之间。

民主制小组的成员之间地位平等，小组的工作目标和决策都是由全体成员集体决定的。这种形式的组内通信路径较多。

层次式小组的组织形式是一名组长领导若干名高级程序员，每名高级程序员领导若干名程序员。组长通常就是项目负责人，负责全组的技术工作，进行任务分配，组织评审。这种组织形式适合于具有层次结构特征的项目的开发。其组内的通信路径数介于主程序员制小组和民主制小组之间。

4.5.5 软件质量管理

软件质量是指反映软件系统或软件产品满足规定或隐含要求的能力的特征和特性全体。软件质量保证是为保证软件系统或软件产品充分满足用户要求的质量而进行的有计划、有组织的活动，其目的是生产具有该质量的软件。

1. 软件质量特性

ISO/IEC 9126 软件质量模型由三个层次组成：第一层是质量特性，第二层是质量子特

性，第三层是度量指标。该模型的质量特性和质量子特性的含义如下。

(1) 功能性：与一组功能及其指定的性质的存在有关的一组属性。功能是指能满足规定或隐含需求的那些功能。

- 适合性：与规定任务能否提供一组功能以及这组功能能否适合有关的软件属性。
- 准确性：与能否得到正确的或相符的结果或效果有关的软件属性。
- 互用性：与同其他指定系统进行交互操作的能力有关的软件属性。
- 依从性：使软件服从有关的标准、约定、法规及类似规定的软件属性。
- 安全性：与避免对程序及数据的非授权故意或意外访问的能力有关的软件属性。

(2) 可靠性：与在规定的一段时间内和规定的条件下，软件维持其性能水平有关的能力。

- 成熟性：与由软件故障引起失效的频度有关的软件属性。
- 容错性：与在软件错误或违反指定接口情况下，维持指定的性能水平的能力有关的软件属性。
- 易恢复性：与在故障发生后重新建立其性能水平并恢复直接受影响数据的能力，以及为达此目的所需的时间有关的软件属性。

(3) 易使用性：与为使用所需的努力和由一组规定的或隐含的用户对如此使用所做的评价有关的一组属性。

- 易理解性：与用户为理解逻辑概念及其应用范围所花的努力有关的软件属性。
- 易学性：与用户为学习其应用(例如操作控制、输入、输出)所需努力有关的软件属性。
- 易操作性：与用户为进行操作或操作控制所需努力有关的软件属性。

(4) 效率：与在规定条件下，软件的性能水平与所用资源量之间的关系有关的一组属性。

- 时间特性：与响应和处理时间以及软件执行其功能时的吞吐量有关的软件属性。
- 资源特性：与软件执行其功能时所使用的资源量以及使用资源的持续时间有关的软件属性。

(5) 可维护性：与进行规定的修改所需努力有关的一组属性。

- 易分析性：与为诊断缺陷或失效原因，或为判定待修改的部分所需努力有关的软件属性。
- 易改变性：与进行修改、调试或适应环境变化所需努力有关的软件属性。
- 稳定性：与修改造成未预料后果的风险有关的软件属性。
- 易测试性：与确认修改软件所需努力有关的软件属性。

(6) 可移植性：与软件从一种环境转移到另一环境的能力有关的一组属性。

- 适应性：与软件无须采用有别于为该软件准备的处理和手段就能适应规定的环境有关的软件属性。
- 易安装性：与在指定环境下安装软件所需努力有关的软件属性。
- 一致性：使软件服从与可移植性有关的标准或约定的软件属性。
- 易替换性：与软件在该软件环境中用来替代指定的其他软件的可能和努力有关的软件属性。

2. 软件质量保证

软件质量保证包括与以下七个主要活动相关的各种任务。

(1) 应用技术方法。

(2) 进行正式的技术评审。

(3) 测试软件。

(4) 标准的实施。

(5) 控制变更。

(6) 度量。

(7) 记录保存和报告。

3. 软件评审

通常，把“质量”理解为“用户满意程度”。为了使用户满意，有如下两个必要条件。

(1) 设计的规格说明书符合用户的要求，这称为设计质量。

(2) 程序按照设计规格说明所规定的情况正确执行，这称为程序质量。

设计质量评审的对象是在需求分析阶段产生的软件需求规格说明、数据需求规格说明，在软件概要设计阶段产生的软件概要设计说明等。

程序质量评审通常是从开发者的角度进行评审，与开发技术直接相关。它是着眼于软件本身的结构、与运行环境的接口以及变更带来的影响而进行的评审活动。

4. 软件容错技术

1) 容错软件的定义

归纳容错软件的定义，有以下四种。

- 规定功能的软件，在一定程度上对自身错误的作用(软件错误)具有屏蔽能力，则称此软件为具有容错功能的软件，即容错软件。
- 规定功能的软件，在一定程度上能从错误状态自动恢复到正常状态，则称为容错软件。
- 规定功能的软件，在因某些原因而发生错误时，仍然能在一定程度上完成预期的功能，则把该软件称为容错软件。
- 规定功能的软件，在一定程度上具有容错能力，则称为容错软件。

2) 容错的一般方法

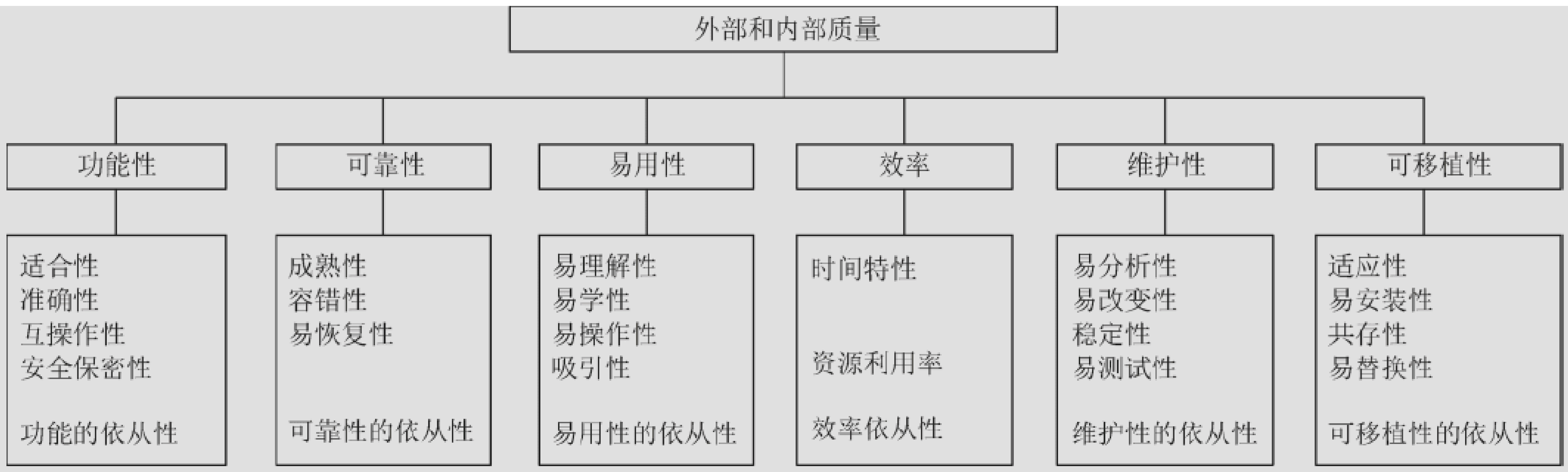
实现容错的主要手段是冗余。冗余是指对于实现系统规定功能是多余的那部分资源，包括硬件、软件、信息和时间。由于加入了这些资源，有可能使系统的可靠性得到较大的提高。通常冗余技术分为四类：结构冗余、信息冗余、时间冗余和冗余附加技术。

真题链接

【例 4-8】在 ISO/IEC 软件质量模型中，可靠性是指在规定的一段时间内和规定的条件下，软件维持在其性能水平的能力；其子特性不包括 (31) 。(2014 年 11 月真题 32)

(31) A. 成熟性　　B. 容错性　　C. 易恢复　　D. 可移植性

【解析】可靠性包含四个子特征: 成熟性、容错性、易恢复和兼容性。



【答案】D

【例 4-9】在屏蔽软件错误的容错系统中，冗余附加技术的构成不包括 (31) 。 (2014 年 5 月真题 31)

(31) A. 关键程序和数据的冗余存储及调用　　B. 冗余备份程序的存储及调用

C. 实现错误检测和错误恢复的程序　　D. 实现容错软件所需的固化程序

【解析】冗余技术又称储备技术，它是利用系统的并联模型来提高系统可靠性的一种手段。冗余附加技术包括：独立设计的相同功能冗余备份程序的存储及调用；实现纠错检测及恢复的程序；为实现容错软件所需的固化程序。

【答案】A

4.5.6　软件配置管理

软件配置管理(Software Configure Management，SCM)用于整个软件工程过程。其目标是标识变更、控制变更、确保变更正确地实现、报告有关变更。SCM 是一组管理整个软件生存期各阶段中变更的活动。

1. 基线

基线是软件生存期中各开发阶段的一个特定点，它的作用是把开发各阶段工作的划分更加明确化，使本来连续的工作在这些点上断开，以便于检查与肯定阶段成果。

2. 软件配置项

软件配置项(SCI)是软件工程中产生的信息项，它是配置管理的基本单位，对已经成为基线的 SCI，虽然可以修改，但必须按照一个特殊的、正式的过程进行评估，确认每一处修改。

3. 版本控制

系统不同版本的表示方法如图 4-8 所示。

4. 变更控制

变更控制是一项最重要的软件配置任务。为有效地实现变更控制，须借助于配置数据库和基线的概念。配置数据库可以分为三类：开发库、受控库和产品库。

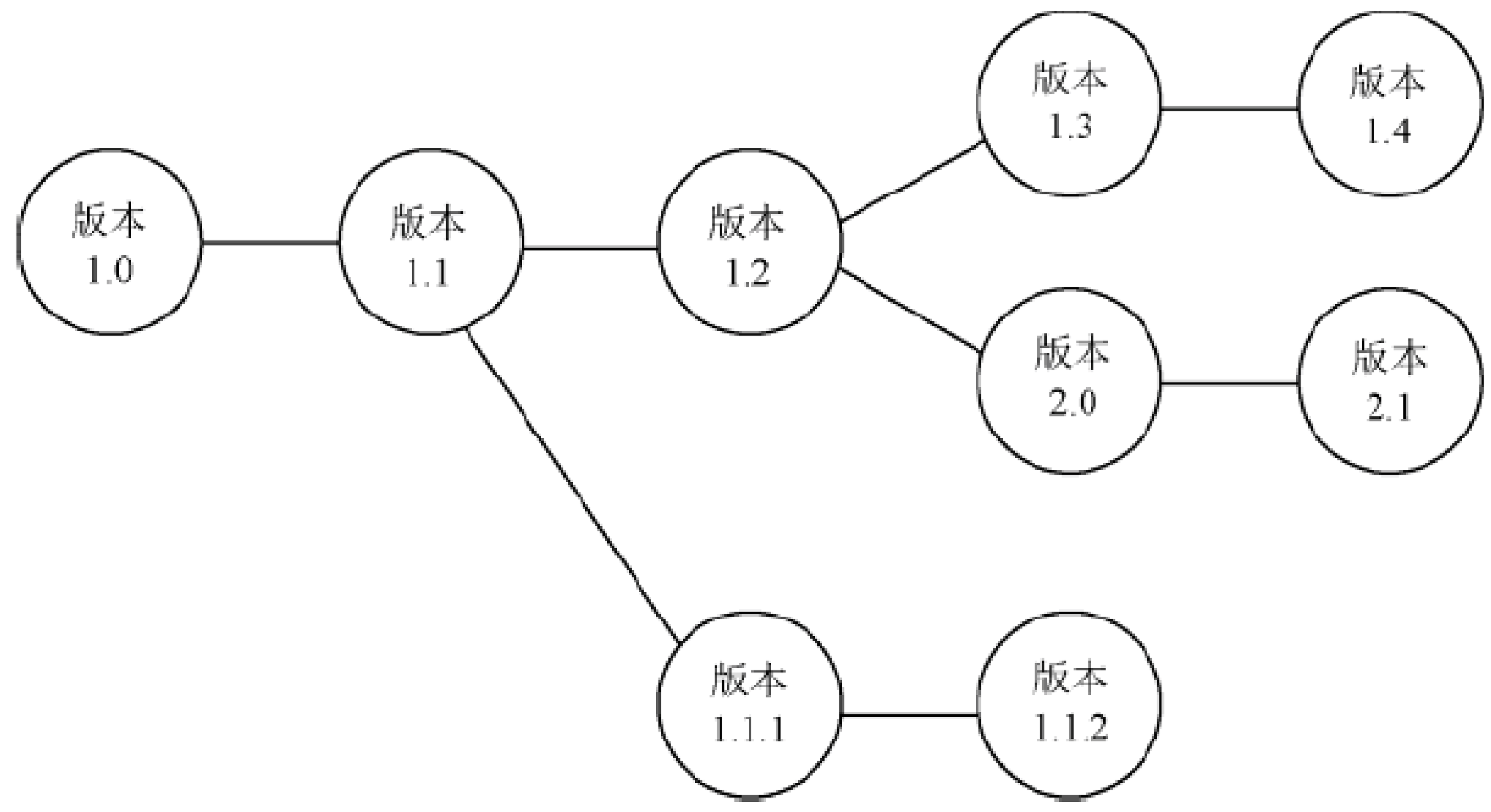


图 4-8　系统不同版本的表示方法

4.6　风 险 管 理

考核说明：本节主要介绍风险管理的相关概念，风险的分类较为重要，考试中经常考到，需要掌握。

4.6.1　软件风险

一般认为软件风险包含两个特性：不确定性和损失。不确定性是指风险可能发生也可能不发生；损失是指如果风险发生，就会产生恶性后果。

项目风险是指预算、进度、人员(聘用职员及组织)、资源、利益相关者、需求等方面的潜在问题以及它们对软件项目的影响。项目复杂度、规模及结构不确定性也属于项目风险因素。项目风险威胁到项目计划。

技术风险是指设计、实现、接口、验证和维护等方面的潜在问题。技术风险威胁到要开发软件的质量及交付时间。

商业风险有市场风险、策略风险、销售风险、管理风险、预算风险。商业风险威胁到要开发软件的生存能力，且常常会危害到项目或产品。

真题链接

【例 4-10】某个项目在开发时采用了不成熟的前沿技术，由此而带来的风险属于 (17) 风险。(2014 年 11 月真题 17)

(17) A. 市场　　B. 技术　　C. 经济　　D. 商业

【解析】技术不成熟属于技术范围的风险。

【答案】B

【例 4-11】项目复杂性、规模和结构的不确定性属于 (19) 风险。(2014 年 5 月真题 19)

(19) A. 项目　　B. 技术　　C. 经济　　D. 商业

【解析】项目风险是指可能导致项目损失的不确定性，美国项目管理大师马克思·怀德曼将其定义为某一事件发生给项目目标带来不利影响的可能性。

【答案】A

4.6.2　风险识别

风险识别是试图系统化地确定对项目计划的威胁。识别风险的一种方法是建立风险条目检查表。该检查表可用来识别下列几种类型中的一些已知风险和可预测风险。

(1) 产品规模。
(2) 商业影响。
(3) 客户特性。
(4) 过程定义。
(5) 开发环境。
(6) 开发技术。
(7) 人员才干及经验。

风险因素包括性能、成本、支持和进度。

4.6.3　风险预测

风险预测又称风险估算，它从两个方面评估一个风险：风险发生的可能性或概率；如果风险发生所产生的后果。

通常，项目计划人员与管理人员、技术人员一起进行以下 4 步风险预测活动。

(1) 建立一个尺度或标准，以反映风险发生的可能性。
(2) 描述风险产生的后果。
(3) 估算风险对项目和产品的影响。
(4) 标注风险预测的整体精确度，以免产生误解。

如果风险真的发生，有 3 个因素可能会影响风险所产生的后果，即风险的本质、范围和时间。整体的风险显露度(Risk Exposure，RE)可由下面的关系确定：

$$\mathrm{RE}=PC$$

其中，P 是风险发生的概率，C 是风险发生时带来的项目成本。

4.6.4　风险评估

在进行风险评估时，建立了如下形式的三元组：

$$(r_i, l_i, x_i)$$

其中，r_i 为风险；l_i 为风险发生的概率；x_i 为风险产生的影响。

一个对风险评估很有用的技术就是定义风险参照水准。对于大多数软件项目来说，成本、进度和性能就是三种典型的风险参照水准。

4.6.5 风险控制

一个有效的策略必须考虑三个问题：①风险避免；②风险监控；③风险管理及意外事件计划。如果软件项目组对于风险采取主动的方法，则避免是最好的策略。

应对风险的最好办法是主动地避免风险，即在风险发生前分析引起风险的原因，然后采取措施，以避免风险的发生。

风险管理策略可以包含在软件项目计划中，或者风险管理步骤也可以组织成一个独立的风险缓解、监控和管理计划(RMMM 计划)。RMMM 计划将所有风险分析工作文档化，并由项目管理者作为整个项目计划中的一部分来使用。

4.7 软 件 度 量

考核说明： 本节主要介绍软件度量分类和软件复杂性度量，McCabe 度量法是重点，也是必考内容。

4.7.1 软件度量分类

软件度量有两种分类方法，第一种分类是将软件度量分为面向规模的度量、面向功能的度量和面向人的度量；第二种分类是将软件度量分为生产率度量、质量度量和技术度量。

1. 面向规模的度量

软件规模通常用程序的代码行(Line of Code，LOC)或千行代码 KLOC 来衡量。计算出软件项目的代码行后，可方便地度量其他的软件属性。面向规模的常用度量公式如下：

LOC 或 KLOC=代码行数或千行代码数

生产率 P= LOC/E，E 为开发的工作量(常用人月数表示)

每行代码平均成本 C=S/LOC，S 为总成本

文档代码比 D= Pe/KLOC，Pe 为文档页数

代码错误率 EQR=N/KLOC，N 为代码中的错误数

2. 面向功能的度量

应用最广泛的面向功能的度量是功能点(Function Point，FP)。功能点是根据软件信息域的特性及复杂性来计算的。信息域的值包括：

(1) 外部输入数(EI)

(2) 外部输出数(EO)

(3) 外部查询数(EQ)

(4) 内部逻辑文件数(ILF)

(5) 外部接口文件数(EIF)

4.7.2　软件复杂性度量

软件复杂性包括程序复杂性和文档复杂性，软件复杂性主要表现为程序的复杂性。程序的复杂性主要指模块内程序的复杂性。程序复杂性主要有以下两种度量方法。

(1)　代码行度量法。度量程序的复杂性，最简单的方法就是统计程序的源代码行数。此方法的基本考虑是统计一个程序的源代码行数，并以源代码行数作为程序复杂性的度量。

(2)　McCabe 度量法。McCabe 复杂性度量又称为环路度量，它认为程序的复杂性很大程度上取决于控制的复杂性。单一的顺序程序结构最为简单，循环和选择所构成的环路越多，程序就越复杂。

根据图论，在一个强连通的有向图 G 中，环的个数 $V(G)$由以下公式给出：

$$V(G)=m-n+2p$$

式中：$V(G)$为有向图 G 中的环路数；m 为图 G 中弧的个数；n 为图 G 中的结点数；p 为图 G 中的强连通分量个数。

真题链接

【例 4-12】如图 4-9 所示的程序流程图中有 (34) 条不同的简单路径，采用 McCabe 度量法计算该程序图的环路复杂性为 (35) 。(2014 年 11 月真题 34、35)

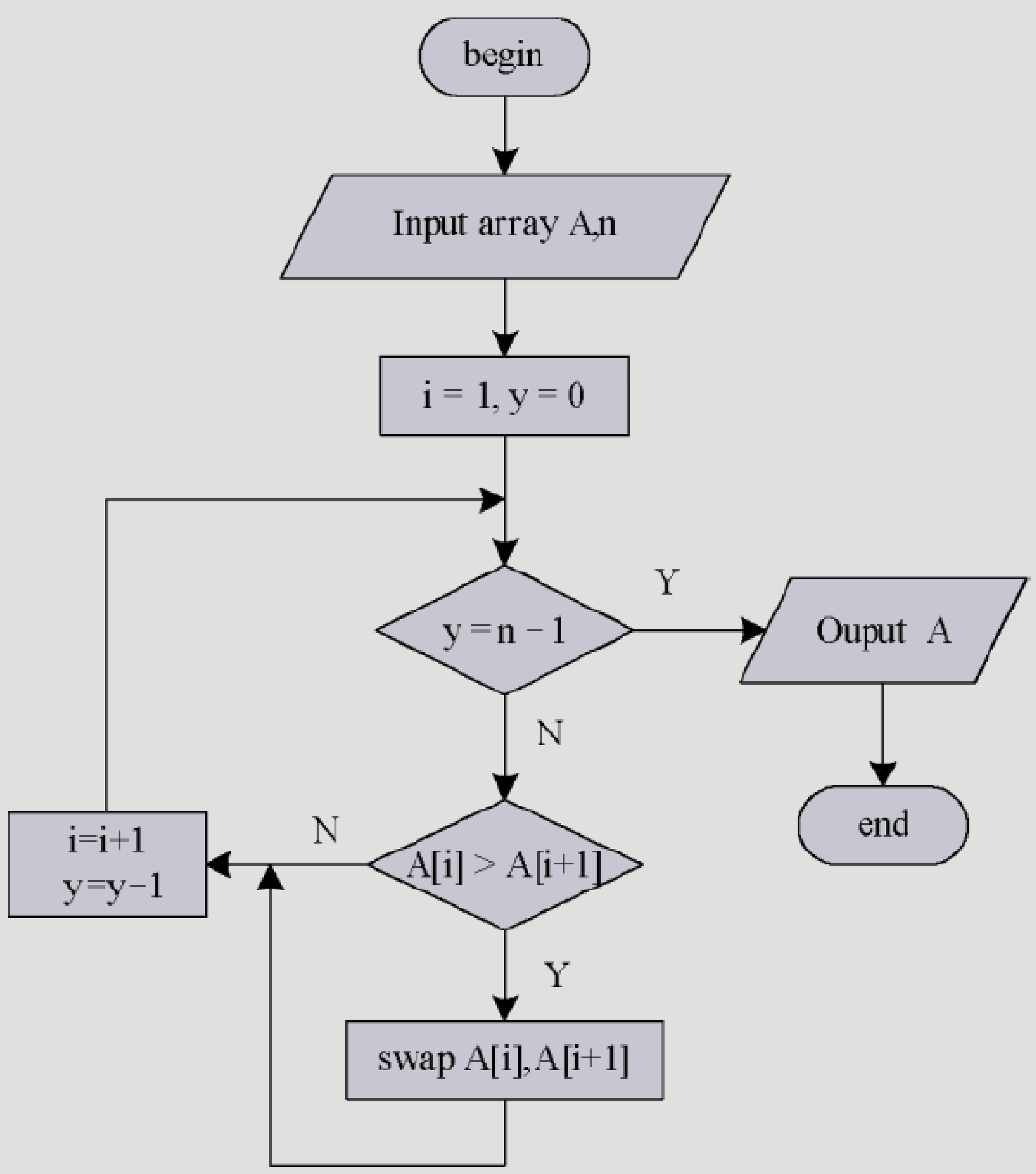


图 4-9　程序流程图

(34)　A. 3　　B. 4　　C. 5　　D. 6

(35)　A. 3　　B. 4　　C. 5　　D. 6

【解析】三条简单路径分别为 begin→y=n−1 成立→output A→end; begin→ y = n−1 不成立→A[i] > A[i+1]不成立→y=n−1 成立→end; begin→ y = n−1 不成立→A[i] > A[i+1]成立→swap A[i], A[i+1] →y=n−1 成立→end。

McCabe 度量法公式为 V(G)=m − n + 2，V(G)是有向图 G 中的环路数，m 是图 G 中的弧数，n 是图 G 中结点数。本题中 m = 10, n = 9, 故 V(G) = 3。

【答案】(34)A；(35)A

【例 4-13】采用 McCabe 度量法计算下列程序图(见图 4-10)的环路复杂性为__(32)__。(2014 年 5 月真题 32)

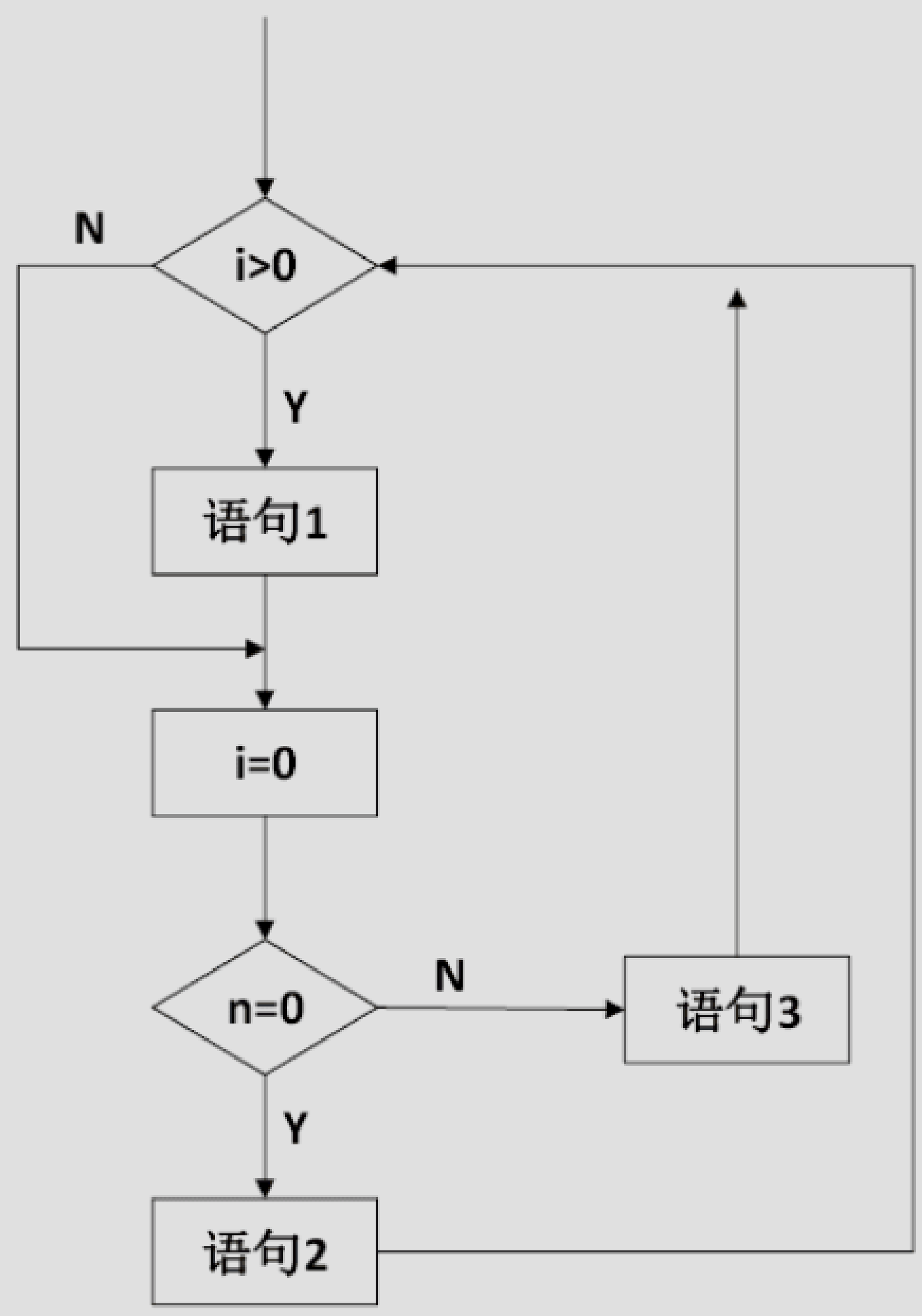


图 4-10 程序流程图

(32) A. 2　　B. 3　　C. 4　　D. 5

【解析】在一个强连通的有向图 G 中，环的个数为 m−n+2p。其中 m 为图中弧的个数，n 为图中的结点数，p 是图中强连通分量的个数。题图为一强连通图，故 p=1，而图中弧的个数为 8，结点数为 6，故环的个数为 8−6+2=4。

【答案】C

4.8 应试加油站

4.8.1 考频统计

通过对近几年考试真题的分析，“软件工程基础知识”模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 4-1 所示。

表 4-1　历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：17、19～20、29、30、32、34、35	增量模型、PERT 图、风险管理策略、统一过程、McCabe 度量法、CMM、能力成熟度模型	6
	下午：无		0
2014 年 5 月	上午：17～19、20～32	增量模型、PERT 图、风险管理策略、统一过程、McCabe 度量法、CMM、能力成熟度模型	7
	下午：无		0
2013 年 11 月	上午：16～19、29～33	数据流图、模块、PERT 图、风险管理策略、软件生命周期模型、增量开发模型、软件质量评审、SEI 能力成熟度模型	9
	下午：无		0
2013 年 5 月	上午：15～17、33、35～36	统一过程、PERT 图、回归测试、数据流图、McCabe 度量法、软件设计质量评审	6
	下午：无		0
2012 年 11 月	上午：16～19、29～33	数据流图、关键路径、模块划分、风险管理、瀑布模型、敏捷开发方法、McCabe 度量法、文档	9
	下午：无		0
2012 年 5 月	上午：17、19、29～30、34～36	数据流图、模块、关键路径、开发人员组织形式、风险管理、增量模型、瀑布模型、Mc Call 软件质量模型、McCabe 度量法、ISO 软件质量模型	7
	下午：无		0

纵观历年试卷，本章知识点是以选择题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量为 6～9 道选择题，所占分值为 6～9 分(约占试卷总分值 75 分中的 8%～12%)。本章试题，难度中等偏难。PERT 图、项目进度管理、软件过程管理、软件质量管理是上午考试的重点，要重点掌握。

4.8.2　解题技巧

【典型题 4-1】__(29)__开发过程模型最不适用于开发初期对软件需求缺乏准确全面认识的情况。(2013 年 11 月真题 29)

(29) A. 瀑布　　B. 演化　　C. 螺旋　　D. 增量

【解析】瀑布模型将软件生命周期的各项活动规定为依固定顺序联接的若干阶段工作，形如瀑布流水，最终得到软件产品，强调早期计划及需求调查；演化模型主要针对事先不能完整定义需求的软件开发，如果所有的产品需求在一开始不完全弄清楚的话，会给总体设计带来困难及削弱产品设计的完整性，并因而影响产品性能的优化及产品的可维护性；螺旋模型是瀑布模型与演化模型相结合，并加入两者所忽略的风险分析所建立的一种软件开发模型；增量模型融合了瀑布模型的基本成分和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。瀑布模型要求明确需求，不适用于开发初期对软件需求缺乏准确全面认识的情况。

【答案】A

【典型题 4-2】__(30)__不是增量式开发的优势。(2013 年 11 月真题 30)

(30) A. 软件可以快速地交付
　　B. 早期的增量作为原型，从而可以加强对系统后续开发需求的理解
　　C. 具有最高优先级的功能首先交付，随着后续的增量不断加入，这就使得更重要的功能得到更多的测试
　　D. 很容易将客户需求划分为多个增量

【解析】增量式开发的优势在于：

① 由于能够在较短的时间内向用户提交一些有用的工作产品，因此能够解决用户的一些急用功能。

② 由于每次只提交用户部分功能，用户有较充分的时间学习和适应新的产品。

③ 对系统的可维护性是一个极大的提高，因为整个系统是由一个个构件集成在一起的，当需求变更时只变更部分部件，而不会响整个系统。

【答案】D

【典型题 4-3】某开发小组预开发一个规模较大、需求较明确的项目。开发小组对项目领域熟悉且该项目与小组开发过的某一项目类似，则适宜采用__(29)__开发过程模型。(2012 年 11 月真题 29)

(29) A. 瀑布　　B. 演化　　C. 螺旋　　D. 喷泉

【解析】 瀑布模型是将软件生存周期各个活动规定为以线性顺序连接的若干阶段的模型，它为软件的开发和维护提供了一种有效的管理模式，适合于软件需求很明确的软件项目的模型。

演化模型在获取一组基本的需求后，通过快速分析构造出该软件的一个初始可运行的版本，然后逐步演化成为最终软件产品，该模型主要是针对在需求不明确或不清楚的情况下，进行项目开发建议采用的，而螺旋模型弥补了以上两种模型的不足，增加了风险分析，每个螺旋周期分为制定计划、风险分析、实施工程和用户评估 4 个工作步骤。

喷泉模型是一种以用户需求为动力，以对象作为驱动的模型，适合于面向对象的开发方法。它克服了瀑布模型不支持软件复用和多项开发活动集成的局限性，使开发过程具有迭代性和无间隙性。

【答案】A

【典型题 4-4】软件开发的增量模型__(29)__。(2012 年 5 月真题 29)

(29) A. 最适用于需求被清晰定义的情况　　B. 是一种能够快速构造可运行产品的好方法
　　C. 最适合于大规模团队开发的项目　　D. 是一种不适用于商业产品的创新模型

【解析】增量模型融合了瀑布模型的基本成分(重复应用)和原型实现的迭代特征，该模型采用随着日程时间的进展而交错的线性序列，每一个线性序列产生软件的一个可发布的“增量”。当使用增量模型时，第 1 个增量往往是核心的产品，即第 1 个增量实现了基本的需求，但很多补充的特征还没有发布。客户对每一个增量的使用和评估都作为下一个增量发布的新特征和功能，这个过程在每一个增量发布后不断重复，直到产生了最终的完善产品。增量模型的特点是引进了增量包的概念，无须等到所有需求都出来，只要某个需求的增量包出来即可进行开发。虽然某个增量包可能还需要进一步适应客户的需求并且更改，但只要这个增量包足够小，其影响对整个项目来说是可以承受的。因此增量模型是一种能够快速构造可运行产品的好方法。

【答案】B

【典型题 4-5】假设某软件公司与客户签订合同开发一个软件系统，系统的功能有较清晰的定义，且

客户对交付时间有严格要求，则该系统的开发最适宜采用__(30)__。(2012 年 5 月真题 30)

(30) A. 瀑布模型　　B. 原型模型　　C. V 模型　　D. 螺旋模型

【解析】瀑布模型是以软件需求完全确定为前提的。瀑布模型将软件的生命周期划分为软件计划、需求分析和定义、软件设计、软件实现、软件测试、软件运行和维护这 6 个阶段，规定了它们自上而下、相互衔接的固定次序，如同瀑布流水逐级下落。瀑布模型有利于大型软件开发过程中人员的组织及管理，有利于软件开发方法和工具的研究与使用，从而提高了大型软件项目开发的质量和效率。本题中，要开发的软件系统功能较为清晰，而且交付时间严格，因此适宜采用瀑布模型。

【答案】A

【典型题 4-6】在进行进度安排时，PERT 图不能清晰地描述__(16)__，但可以给出哪些任务完成后才能开始另一些任务。某项目 X 包含任务 A、B、…、J，其中 PERT 图如图 4-11 所示(A=1 表示任务 A 的持续时间是 1 天)，则项目 X 的关键路径是__(17)__。(2013 年 5 月真题 16～17)

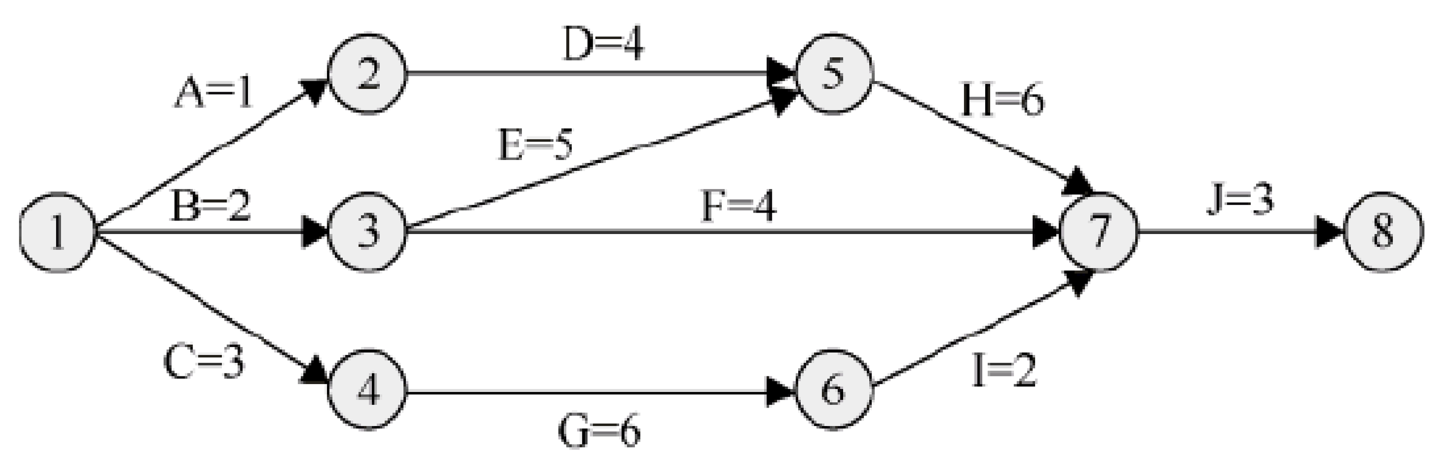


图 4-11　PEBT 图

(16) A. 每个任务从何时开始　　B. 每个任务到何时结束

C. 各任务之间的并行情况　　D. 各任务之间的依赖关系

(17) A. A-D-H-J　　B. B-E-H-J　　C. B-F-J　　D. C-G-I-J

【解析】PERT 图是一个有向图，箭头表示任务，可以标上完成该任务所需的时间；箭头指向节点表示流入节点的任务的结束，并开始流出节点的任务，节点表示事件。PERT 图不仅给出了每个任务的开始时间、结束时间和完成该任务所需的时间，还给出了任务之间的关系，即哪些任务完成后才能开始另外一些任务，但 PERT 图不能清晰地描述各个任务之间的并行关系。

图中用时最长的路径为关键路径，其关键路径为 B-E-H-J。

【答案】(16)C；(17)B

【典型题 4-7】图 4-12 是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，则里程碑__(17)__在关键路径上。若在实际项目进展中，活动 AD 在活动 AC 开始 3 天后才开始，而完成活动 DG 过程中，由于有临时事件发生，实际需要 15 天才能完成，则完成该项目的最短时间比原计划多了__(18)__天。(2013 年 11 月真题 17～18)

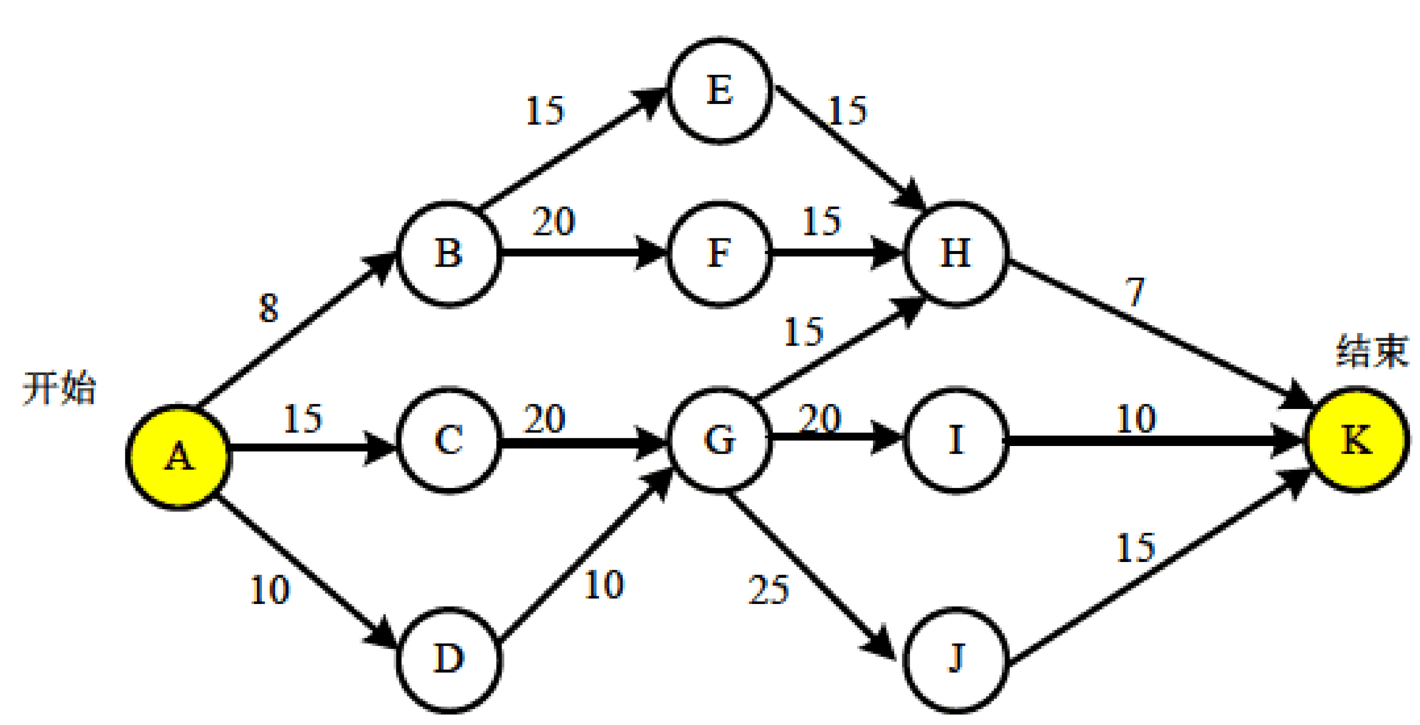


图 4-12　某软件项目的活动图

(17) A. B　　B. C　　C. D　　D. I

(18) A. 8　　B. 3　　C. 5　　D. 6

【解析】用时最长的路径为关键路径，为 A→C→G→J→K，所需时间为 75。任务的松弛时间表示在不影响整个工期的前提下，完成该任务有多少机动余地。活动 AD 的最早开始时间为 10+3=13，DG 所花时间为 10+15=25，A→D→G→J→K 时间为 78，所以说最短时间比原计划多了 78−75=3 天。

【答案】(17)B；(18)B

【典型题 4-8】针对“关键职员在项目未完成时就跳槽”的风险，最不合适的风险管理策略是<u>(19)</u>。(2013 年 11 月真题 19)

(19) A. 对每一个关键性的技术人员，要培养后备人员

B. 建立项目组，以使大家都了解有关开发活动的信息

C. 临时招聘具有相关能力的新职员

D. 对所有工作组织细致的评审

【解析】风险管理策略主要有：①避免风险策略；②控制风险策略；③分散与中和风险策略；④转移风险策略；等等。根据题意：选项 A、B 是分散与中和风险策略，选项 D 是控制风险策略，所以本题答案选 C。

【答案】C

【典型题 4-9】某项目软件的活动图如图 4-13 所示。图中顶点表示项目里程碑，连接顶点的边表示包含的活动，则里程碑<u>(16)</u>在关键路径上，活动 FG 的松弛时间为<u>(17)</u>。(2012 年 11 月真题 16、17)

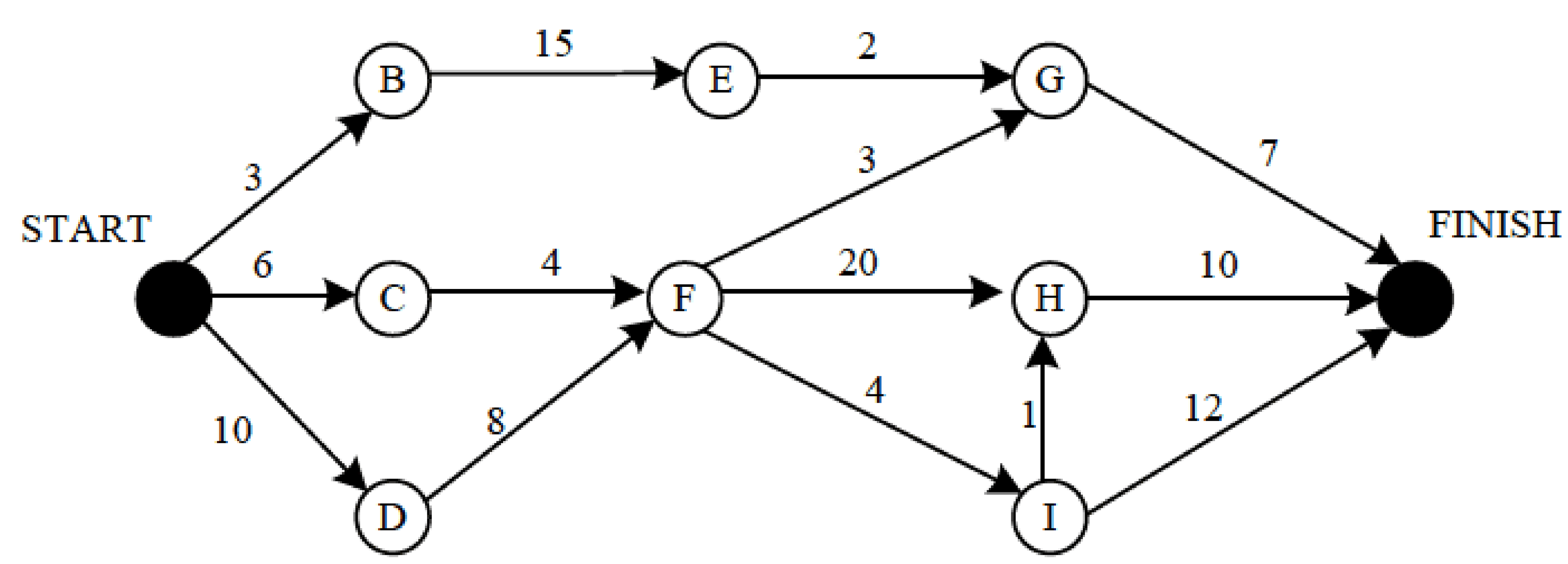


图 4-13　活动图

(16) A. B　　B. C　　C. D　　D. I

(17) A. 19　　B. 20　　C. 21　　D. 24

【解析】 用时最长的路径为关键路径，为 START→D→F→H→FINISH，所需时间为 48。任务的松弛时间表示在不影响整个工期的前提下，完成该任务有多少机动余地。活动 FG 的最早开始时间为 18，最迟开始时间为 48−7−3=38，因此松弛时间为 38−18=20。

【答案】(16)C；(17)B

【典型题 4-10】定义风险参照水准是<u>(19)</u>活动常用的技术。(2012 年 11 月真题 19)

(19) A. 风险识别　　B. 风险预测　　C. 风险评估　　D. 风险控制

【解析】一个对风险评估很有用的技术就是定义风险参照水准。对于大多数软件项目来说，成本、进度和性能就是 3 种典型的风险参照水准。

【答案】C

【典型题 4-11】图 4-14 是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的值表示完成活动所需要的时间，则<u>(17)</u>在关键路径上。(2012 年 5 月真题 17)

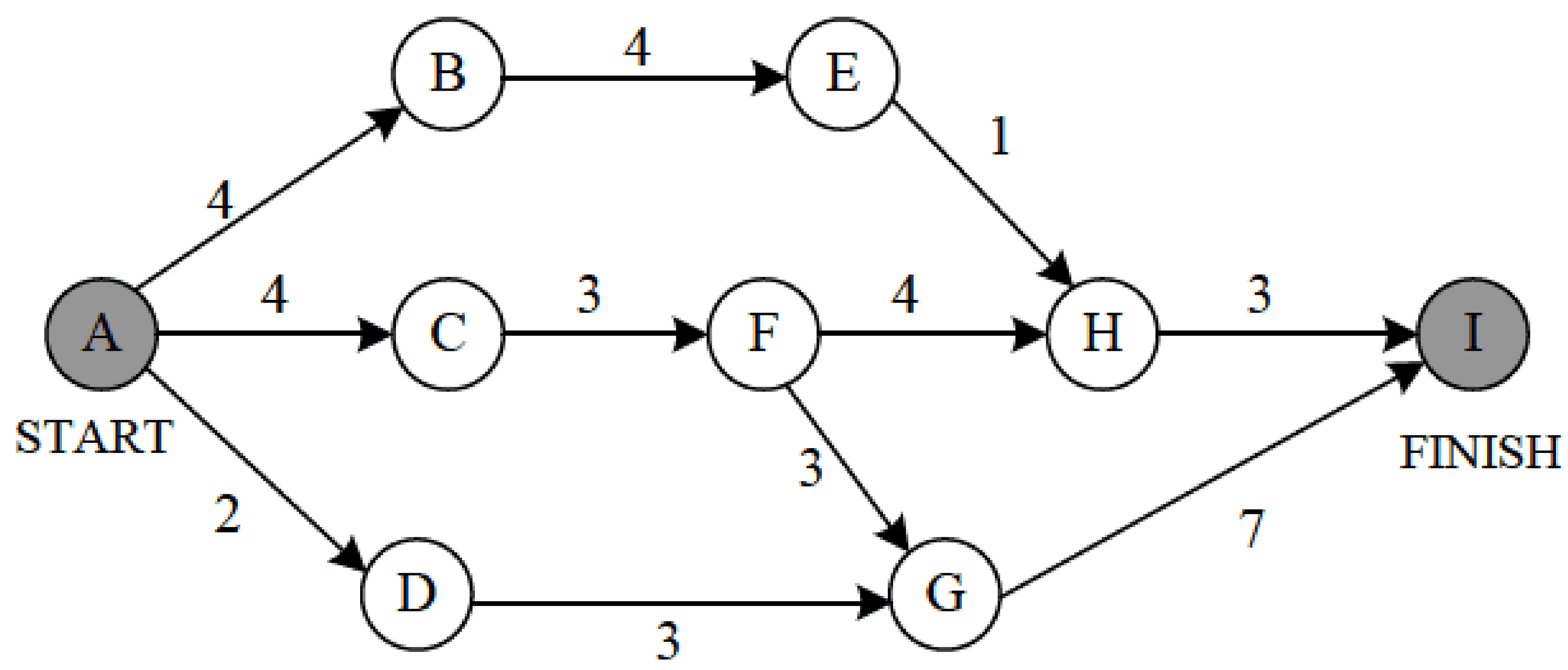


图 4-14　活动图

(17) A. B　　　　B. C　　　　C. D　　　　D. H

【解析】 这种题基本上是必考题，关键路径可以简单地记为：完成活动需要时间最长的路径。本题的关键路径为 A→C→F→G→I，所需时间为 17，很显然 C 在关键路径上。

【答案】 B

【典型题 4-12】 __(18)__ 最不适宜采用无主程序员组的开发人员组织形式。(2012 年 5 月真题 18)

(18) A. 项目开发人数少(如 3～4 人)的项目　　B. 采用新技术的项目

C. 大规模项目　　D. 确定性较小的项目

【解析】 无主程序员组中的成员相互平等，工作目标和决策都由全体成员民主讨论。这种组有利于发挥每个人的积极性，但往往职责不明，出了问题谁也不负责，而且不利于与外界的联系。很显然，大规模的项目不适合采用这种人员组织形式。

【答案】 C

【典型题 4-13】 若软件项目组对风险采用主动的控制方法，则 __(19)__ 是最好的风险控制策略。(2012 年 5 月真题 19)

(19) A. 风险避免　　B. 风险监控

C. 风险消除　　D. 风险管理及意外事件计划

【解析】 如果软件项目组对于风险采取主动的方法，则避免永远是最好的策略。这可以通过建立一个风险缓解计划来达到。例如，频繁的人员流动被标注为一个项目风险，基于以往的历史和管理经验，人员流动的概率为 70%，被预测为对于项目成本及进度有严重的影响。为了缓解这个风险，项目管理者必须建立一个策略来降低人员流动。

【答案】 A

【典型题 4-14】 统一过程是一种“用例和风险驱动，以架构为中心，迭代并且增量”的开发过程，定义了不同阶段及其制品，其中精华阶段关注 __(15)__ 。(2013 年 5 月真题 15)

(15) A. 项目的初创活动　　B. 需求分析和架构演进

C. 系统的构建，产生实现模型　　D. 软件提交方面的工作，产生软件增量

【解析】 精化阶段的目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。精化阶段结束是第二个重要的里程碑：生命周期结构(Lifecycle Architecture)里程碑。生命周期结构里程碑为系统的结构建立了管理基准并使项目小组能够在构建阶段中进行衡量。此刻，要检验详细的系统目标和范围、结构的选择以及主要风险的解决方案。

【答案】 B

【典型题 4-15】 SEI 能力成熟度模型(SEI CMM)把软件开发企业分为 5 个成熟度级别，其中 __(32)__ 重

点关注产品和过程质量。(2013 年 11 月真题 32)

(32) A. 级别 2：重复级　　B. 级别 3：确定级
C. 级别 4：管理级　　D. 级别 5：优化级

【解析】级别 1：初始级。处于这个最低级的组织，基本上没有健全的软件工程管理制度。每件事情都以特殊的方法来做。如果一个特定的工程碰巧由一个有能力的管理员和一个优秀的软件开发组来做，则这个工程可能是成功的。然而通常的情况是，由于缺乏健全的总体管理和详细计划，时间和费用经常超支。结果，大多数的行动只是应付危机，而非事先计划好的任务。处于成熟度等级第 1 级的组织，由于软件过程完全取决于当前的人员配备，所以具有不可预测性，人员变化了，过程也跟着变化。结果，要精确地预测产品的开发时间和费用之类重要的项目，是不可能的。

级别 2：重复级。在这一级，有些基本的软件项目的管理行为、设计和管理技术是基于相似产品中的经验，故称为“可重复”。在这一级采取了一定措施，这些措施是实现一个完备过程所必不可少的第一步。典型的措施包括仔细地跟踪费用和进度。不像在第 1 级那样，在危机状态下行动，管理人员在问题出现时便可发现，并立即采取修正行动，以防它们变成危机。关键的一点是，如没有这些措施，要在问题变得无法收拾前发现它们是不可能的。在一个项目中采取的措施也可用来为未来的项目拟定实现的期限和费用计划。

级别 3：定义级。在第 3 级，已为软件生产的过程编制了完整的文档。软件过程的管理方面和技术方面都明确地做了定义，并按需要不断地改进过程，而且采用评审的办法来保证软件的质量。在这一级，可引用 CASE 环境来进一步提高质量和产生率。而在第 1 级过程中，“高技术”只会使这一危机驱动的过程更混乱。

级别 4：管理级。一个处于第 4 级的公司对每个项目都设定质量和生产目标。这两个量将被不断地测量，当偏离目标太多时，就采取行动来修正。利用统计质量控制，管理部门能区分出随机偏离和有深刻含义的质量或生产目标的偏离(统计质量控制措施的一个简单例子是每千行代码的错误率。相应的目标就是随时间推移减少这个量)。

级别 5：优化级。1 个第 5 级组织的目标是连续地改进软件过程。这样的组织使用统计质量和过程控制技术作为指导。从各个方面中获得的知识将被运用在以后的项目中，从而使软件过程融入了正反馈循环，使生产率和质量得到稳步的改进。

【答案】C

【典型题 4-16】敏捷开发方法中，<u>(30)</u>认为每一种不同的项目都需要一套不同的策略、约定和方法论。(2012 年 11 月真题 30)

(30) A. 极限编程(XP)　　B. 水晶法(Crystal)
C. 并列争球法(Scrum)　　D. 自适应软件开发(ASD)

【解析】敏捷开发方法有很多，每一种方法基于一套原则。极限编程(XP)是一种轻量级、高效、低风险、可预测的、科学的软件开发方式，由 4 个价值观、5 个原则、12 个最佳实践与行为等 4 部分组成。水晶法(Crystal)认为每一个不同的项目都需要一套不同的策略、约定和方法论。并列争球法(Scrum)使用迭代的方法，其中把每 30 天一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品。自适应软件开发(ASD)包含 6 个基本的原则：有一个使命作为指导；特征被视为客户价值的关键点；过程中的等待是很重要的；变化不被视为改正，而是被视为对软件开发实际情况的调整；确定的交付时间迫使开发人员认真考虑每一个生产版本的关键需求；风险也包含其中。

【答案】B

【典型题 4-17】软件的复杂性主要体现在程序的复杂性。__(30)__是度量软件复杂性的一个主要参数。若采用 McCabe 度量法计算环路复杂性，则对于图 4-15 所示的程序图，其环路复杂度为__(31)__。(2013 年 5 月真题 30～31)

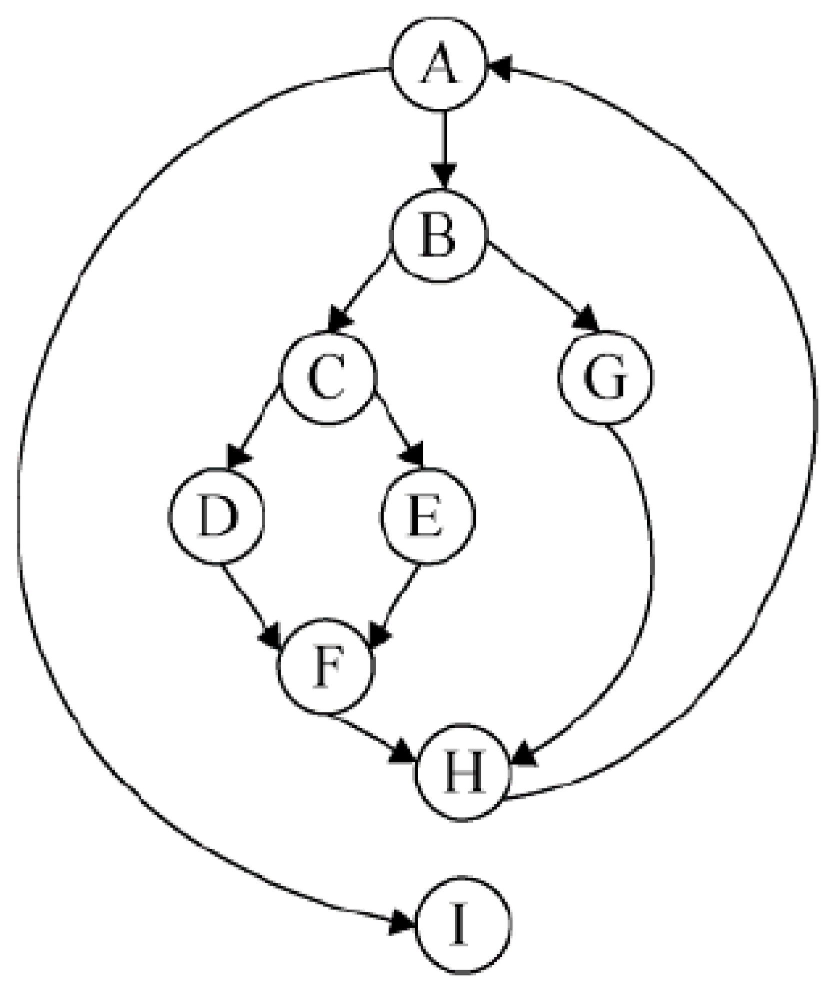


图 4-15　程序图

(30) A. 代码行数　　B. 常量的数量　　C. 变量的数量　　D.调用的库函数的数量

(31) A. 2　　B. 3　　C. 4　　D. 5

【解析】代码行数度量法以程序的总代码行数作为程序复杂性的度量值。这种度量方法有一个重要的隐含假定是：书写错误和语法错误在全部错误中占主导地位。

McCabe 度量法是一种基于程序控制流的复杂性度量方法。采用这种方法要先画出程序图，然后采用公式 $V(G)=m-n+2$ 计算环路复杂度。其中，m 是图 G 中弧的个数，n 是图 G 中的结点数。图中结点数为 9，边数为 11，所以环路复杂度为 11−9+2=4。

【答案】(30)A；(31)C

【典型题 4-18】__(32)__不属于软件设计质量评审。(2013 年 5 月真题 32)

(32)　A. 功能与模块之间的对应关系　　B. 软件规格说明是否符合用户的需求
　　　C. 软件是否具有可测试性　　D. 软件是否具有良好的可靠性

【解析】软件质量评审包括以下内容：①评价软件的规格说明是否合乎用户的要求，即总体设计思想和设计方针是否正确。②评审可靠性，即是否能避免输入异常(错误或超载等)、硬件失效及软件失效所产生的失效。③评审保密措施实现情况。④评审操作特性实施情况。⑤评审性能实现情况。⑥评审软件是否具有可修改性、可扩充性、可互换性和可移植性。⑦评审软件是否具有可测试性。⑧评审软件是否具有可复用性。

因此 A 选项不属于软件质量评审。

【答案】A

【典型题 4-19】在对软件系统进行评价时，需要从信息系统的组成部分、评价对象和经济学角度出发进行综合考虑以建立起一套指标体系理论架构。从信息系统评价对象出发，对于用户方来说，他们所关心的是__(34)__。(2013 年 5 月真题 34)

(34) A. 用户需求和运行质量　　B. 系统外部环境
　　C. 系统内部结构　　D. 系统质量和技术水平

【解析】在对软件系统进行评价时，从系统评价对象出发，用户所关心的是他们的需求和系统的运行质量。

【答案】A

【典型题 4-20】在对程序质量进行评审时，模块结构是一个重要的评审项，评审内容中不包括__(31)__。(2013 年 11 月真题 31)

(31) A. 数据结构　　B. 数据流结构
C. 控制流结构　　D. 模块结构与功能结构之间的对应关系

【解析】程序质量评审通常是从开发者的角度进行评审，直接与开发技术有关。它着眼于软件本身的结构、与运行环境的接口、变更带来的影响而进行的评审活动，模块结构主要包括以下几项内容。

① 控制流结构：规定了处理模块与处理模块之间的流程关系。检查处理模块之间的控制转移关系与控制转移形式(调用方式)。

② 数据流结构：规定了数据模块是如何被处理模块加工的流程关系。检查处理模块与数据模块之间的对应关系；处理模块与数据模块之间的存取关系，如建立、删除、查询、修改等。

③ 模块结构与功能结构之间的对应关系：包括功能结构与控制流结构的对应关系；功能结构与数据流结构的对应关系；每个模块的定义(包括功能、输入与输出数据)。

【答案】A

【典型题 4-21】系统可维护性的评价指标不包括__(33)__。(2013 年 11 月真题 33)

(33) A. 可理解性　　B. 可测试性
C. 可移植性　　D. 可修改性

【解析】系统可维护性的评价指标主要包括：可理解性、可测试性、可修改性和维护工具。所以答案选 C。

【答案】C

【典型题 4-22】采用 McCabe 度量法计算图 4-16 所示环路的复杂性为__(31)__。(2012 年 11 月真题 31)

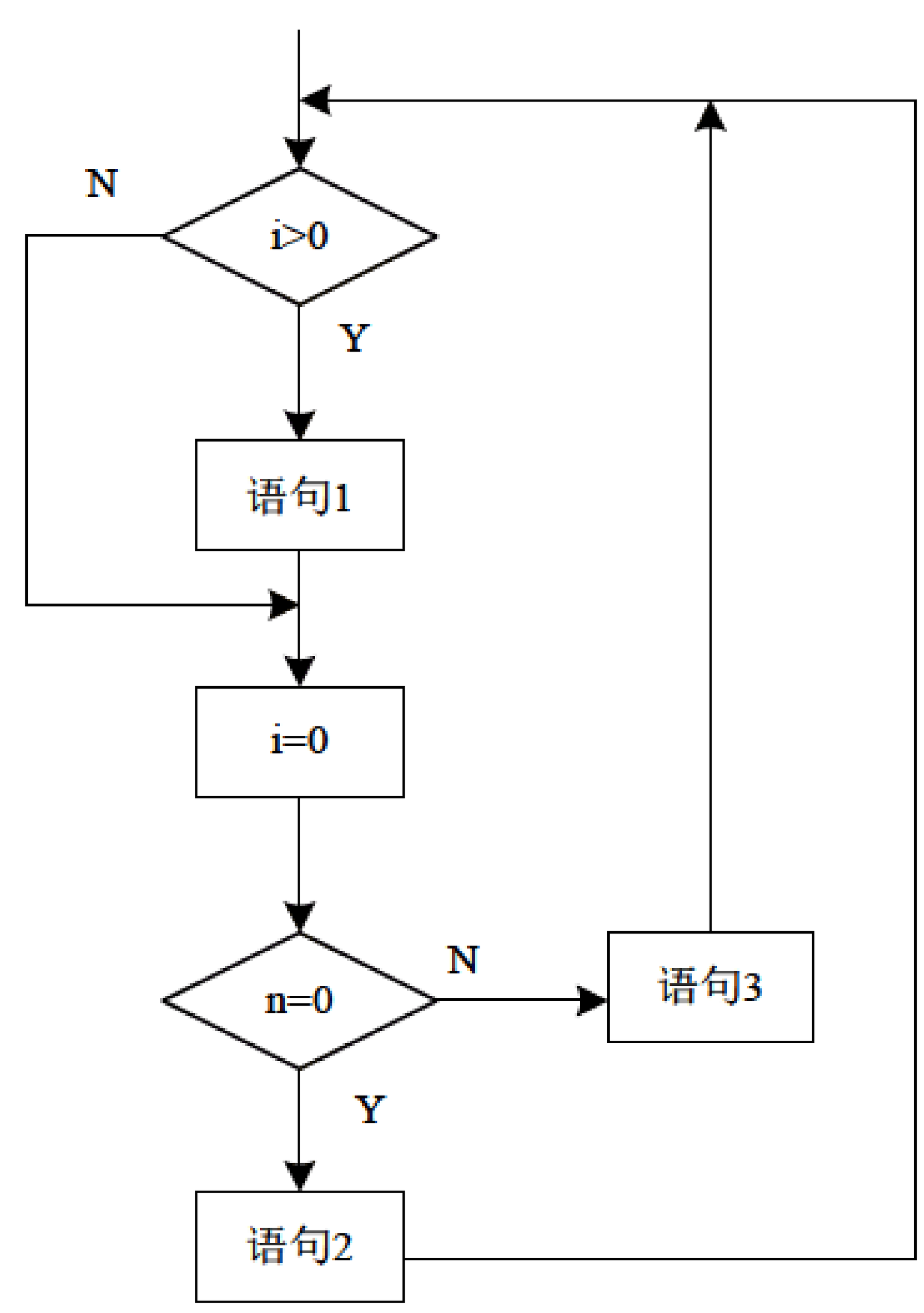


图 4-16　程序流程图

(31) A. 2　　B. 3　　C. 4　　D. 5

【解析】McCabe 度量法是一种基于程序控制流的复杂性度量方法，该方法认为程序的复杂性很大程度上取决于控制的复杂性。首先根据程序画出程序图，然后基于图论用图的环路数来度量程序复杂性，即 $V(G)=m-n+2p$，其中 m、n 和 p 分别表示图 G 中弧的个数、顶点的个数和强连通分量数。由图 4-16 可知，m=8，n=6，p=1，根据上述公式可得，其环路复杂性为 8-6+2=4。

【答案】C

【典型题 4-23】在屏蔽软件错误的容错系统中，冗余附加技术的构成不包括__(32)__。(2012 年 11 月真题 32)

(32) A. 关键程序和数据的冗余存储及调用　　B. 冗余备份程序的存储及调用
C. 实现错误检测和错误恢复的程序　　D. 实现容错软件所需的固化程序

【解析】冗余附加技术是指为实现冗余技术所需的资源和技术，包括程序、指令、数据、存放和调动它们的空间和通道等。在屏蔽硬件错误的容错技术中，冗余附加技术包括关键程序和数据的冗余存储及调用，检测、表决、切换、重构、纠错和复算的实现。在屏蔽软件错误的容错系统中，冗余附加技术的构成包括冗余备份程序的存储及调用、实现错误检测和错误恢复的程序、实现容错软件所需的固化程序。

【答案】A

【典型题 4-24】Mc Call 软件质量模型从软件产品的运行、修正和转移三个方面确定了 11 个质量特性，其中__(32)__不属于产品运行方面的质量特性。(2012 年 5 月真题 32)

(32) A. 正确性　　B. 可靠性　　C. 效率　　D. 灵活性

【解析】产品运行方面的质量特性有：正确性、可靠性、易使用性、效率、完整性。灵活性则属于产品修正方面的特性。

【答案】D

【典型题 4-25】采用 McCabe 度量法计算图 4-17 所示程序图的环路复杂性为__(33)__。(2012 年 5 月真题 33)

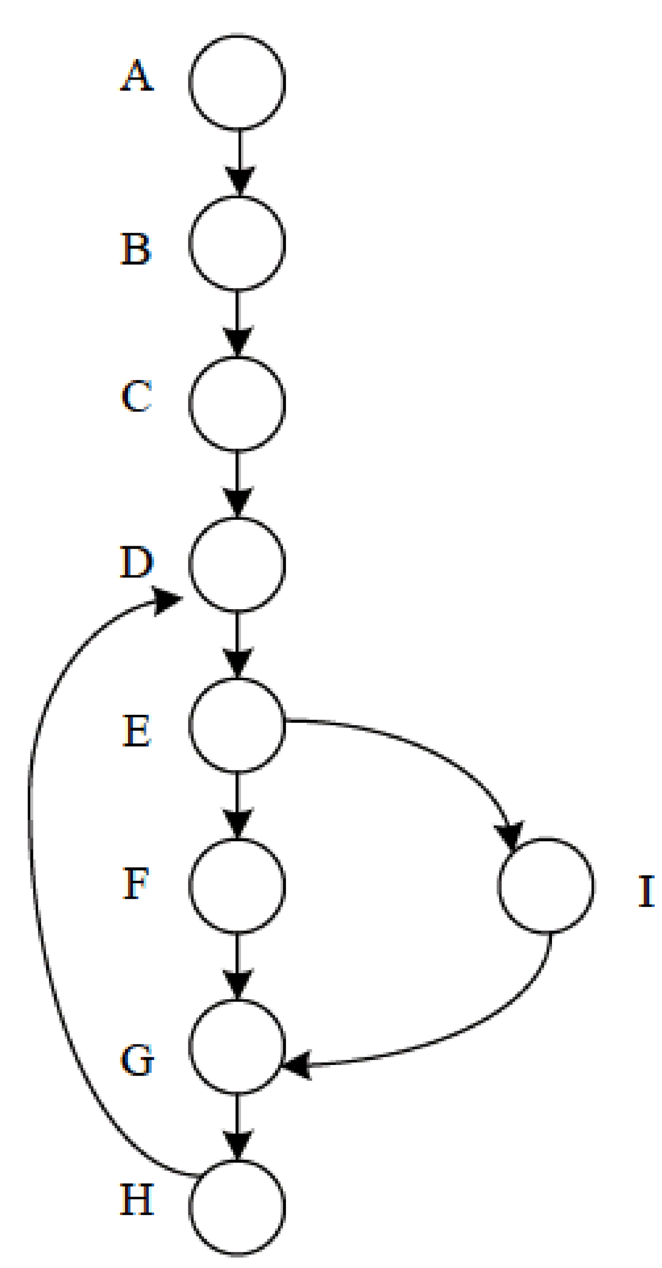


图 4-17　程序图

(33) A. 2　　B. 3　　C. 4　　D. 5

【解析】对于有向图，复杂性度量值为 $m-n+2$，其中 m 是图中弧的个数，n 是图中的节点数。本题中

弧的个数为 10，节点数为 9，因此复杂度为 10−9+2=3。

【答案】 B

【典型题 4-26】 根据 ISO/IEC 9126 软件质量模型中对软件质量特性的定义，可维护性质量特性的_(36)_子特性是指为确认修改软件所需努力有关的软件属性。(2012 年 5 月真题 36)

(36) A. 易测试性　　B. 易分析性

C. 稳定性　　D. 改变性

【解析】 可维护性的子特性有：易分析性、易改变性、稳定性和易测试性。

① 易分析性：与为诊断缺陷或失效原因，或为判定待修改的部分所需努力有关的软件属性。

② 易改变性：与进行修改、调试或适应环境变化所需努力有关的软件属性。

③ 稳定性：与修改造成未预料后果的风险有关的软件属性。

④ 易测试性：与确认修改软件所需努力有关的软件属性。

【答案】 A

4.9 过关习题

1. 以下关于喷泉模型的叙述中，不正确的是______。

A. 喷泉模型是以对象作为驱动的模型，适合于面向对象的开发方法

B. 喷泉模型克服了瀑布模型不支持软件重用和多项开发活动集成的局限性

C. 模型中的开发活动常常需要重复多次，在迭代过程中不断地完善软件系统

D. 各开发活动(如分析、设计和编码)之间存在明显的边界

2. 若全面采用新技术开发一个大学记账系统，以替换原有的系统，则宜选择______进行开发。

A. 瀑布模型　　B. 演化模型　　C. 螺旋模型　　D. 原型模型

3. 为了有效地捕获系统需求，应采用______。

A. 瀑布模型　　B. V 模型　　C. 原型模型　　D. 螺旋模型

4. 某项目组拟开发一个大规模系统，且具备了相关领域及类似规模系统的开发经验。下列过程模型中，______最适合开发此项目。

A. 原型模型　　B. 瀑布模型　　C. V 模型　　D. 螺旋模型

5. 下列关于项目估算方法的叙述不正确的是______。

A. 专家判断方法受到专家经验的主观性影响

B. 启发式方法(如 COCOMO 模型)的参数难以确定

C. 机器学习方法难以描述训练数据的特征和确定其相似性

D. 结合上述三种方法可以得到精确的估算结果

6. 以下关于风险管理的叙述中，不正确的是______。

A. 仅根据风险产生的后果来对风险排优先级

B. 可以通过改变系统性能或功能需求来避免某些风险

C. 不可能去除所有风险，但可以通过采取行动来降低或减轻风险

D. 在项目开发过程中，需要定期地评估和管理风险

7. 下列关于风险的叙述不正确的是：风险是指______。

A. 可能发生的事件　　B. 一定会发生的事件

C. 会带来损失的事件　　D. 可能对其进行干预，以减少损失的事件

8. 在软件开发过程中进行风险分析时，______活动的目的是辅助项目组建立处理风险的策略，有效的策略应考虑风险避免、风险监控、风险管理及意外事件计划。

A. 风险识别　　B. 风险预测　　C. 风险评估　　D. 风险控制

9. 甘特(Gantt)图不能______。

A. 作为项目进度管理的一个工具

B. 清晰地描述每个任务的开始和截止时间

C. 清晰地获得任务并行的信息

D. 清晰地获得各任务之间的依赖关系

10. 图 4-18 是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的值表示完成活动所需要的时间，则关键路径长度为______。

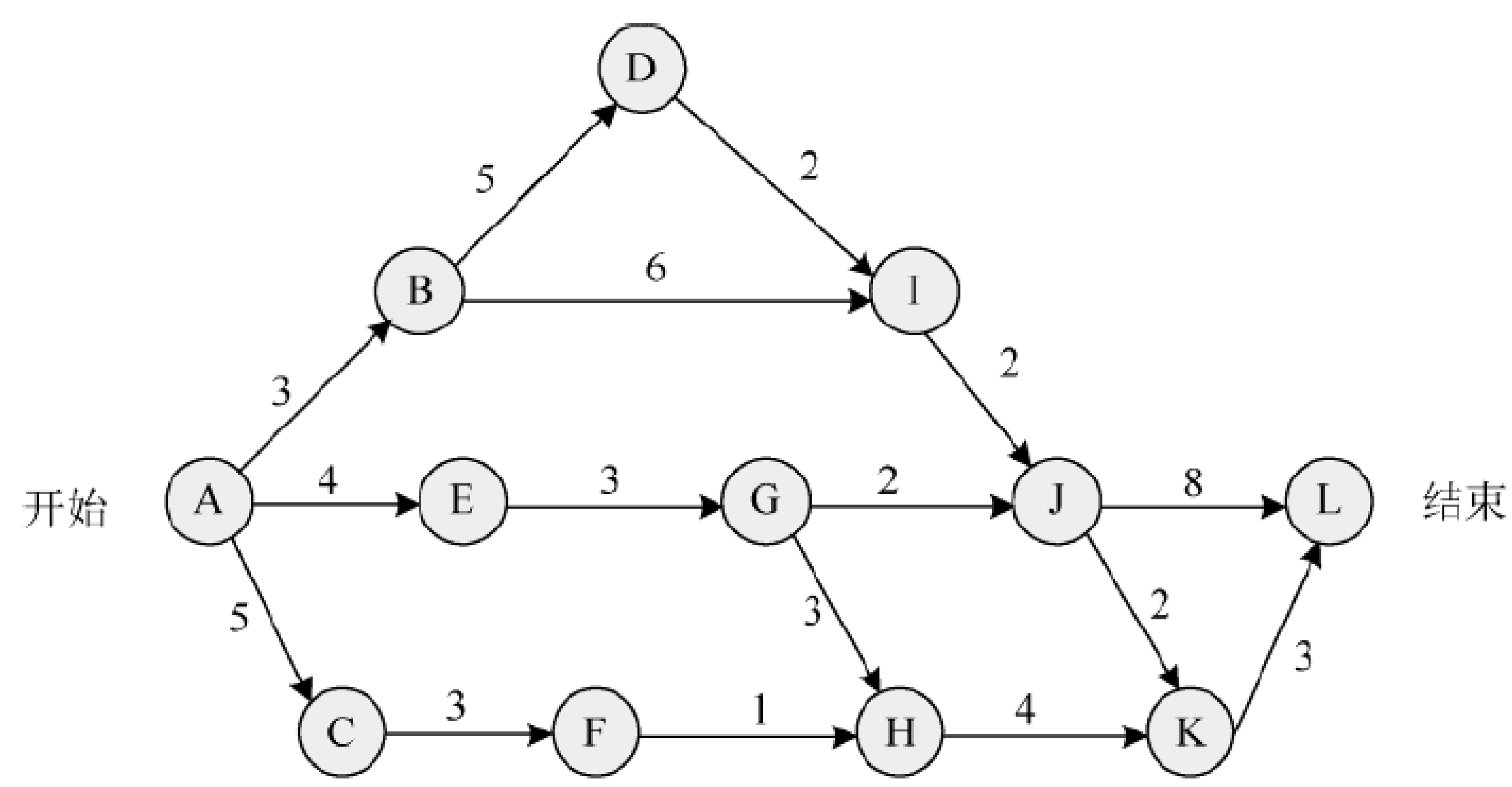


图 4-18　软件项目活动图

A. 20　　B. 19　　C. 17　　D. 16

11. 使用 PERT 图进行进度安排，不能清晰地描述 (1) ，但可以给出哪些任务完成后才能开始另一些任务。图 4-19 所示工程从 A 到 K 的关键路径是 (2) (图中省略了任务的开始和结束时刻)。

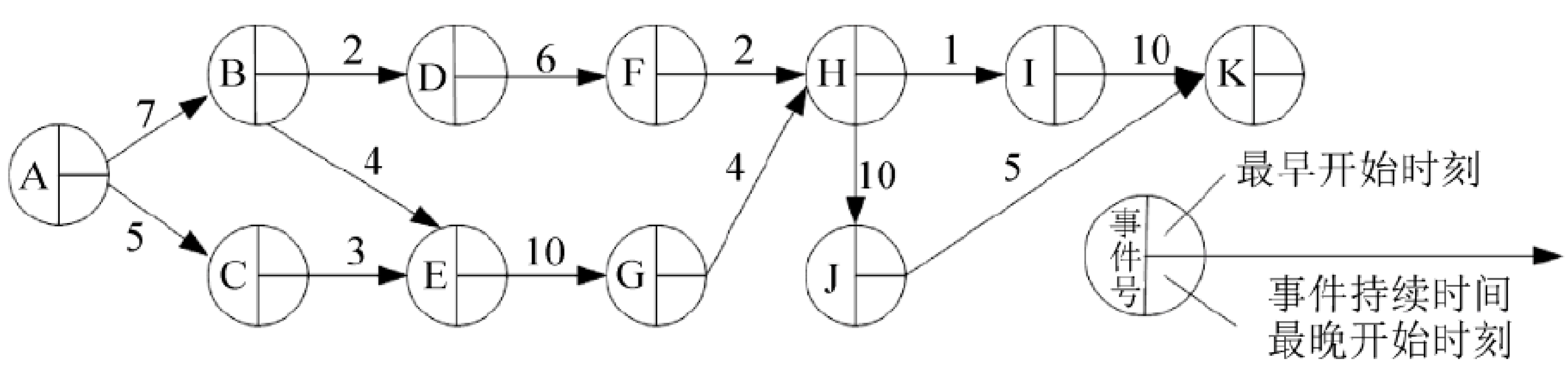


图 4-19　PERT 图

(1) A. 每个任务从何时开始　　B. 每个任务到何时结束

C. 各任务之间的并行情况　　D. 各任务之间的依赖关系

(2) A. ABEGHIK　　B. ABEGHJK　　C. ACEGHIK　　D. ACEGHJK

12. 包含 8 个成员的开发小组的沟通路径最多有______条。

A. 28　　B. 32　　C. 56　　D. 64

13. 关于过程改进，以下叙述中不正确的是______。

A. 软件质量依赖于软件开发过程的质量，其中个人因素占主导作用

B. 要使过程改进有效，需要制定过程改进目标

C. 要使过程改进有效，需要进行培训

D. CMMI 是一种过程改进模型，仅支持阶段性过程改进而不支持连续性过程改进

14. 以下关于过程改进的叙述中，错误的是______。

A. 过程能力成熟度模型基于这样的理念：改进过程将改进产品，尤其是软件产品

B. 软件过程改进框架包括评估、计划、改进和监控四个部分

C. 软件过程改进不是一次性的，需要反复进行

D. 在评估后要把发现的问题转化为软件过程改进计划

15. 敏捷开发方法 XP 是一种轻量级、高效、低风险、柔性、可预测的、科学的软件开发方法，其特性包含在 12 个最佳实践中。系统的设计要能够尽可能早交付，属于______最佳实践。

A. 隐喻　　B. 重构　　C. 小型发布　　D. 持续集成

16. 将每个用户的数据和其他用户的数据隔离开，是考虑了软件的______质量特性。

A. 功能性　　B. 可靠性　　C. 可维护性　　D. 易使用性

17. 软件产品的可靠度并不取决于______。

A. 潜在错误的数量　　B. 潜在错误的位置

C. 软件产品的使用方法　　D. 软件产品的开发方式

18. 软件______是指一个系统在给定时间间隔内和给定条件下无失效运行的概率。

A. 可靠性　　B. 可用性　　C. 可维护性　　D. 可伸缩性

19. 软件系统的可维护性评价指标不包括______。

A. 可理解性　　B. 可测试性　　C. 扩展性　　D. 可修改性

20. 软件复杂性度量的参数不包括______。

A. 软件的规模　　B. 开发小组的规模　　C. 软件的难度　　D. 软件的结构

21. 根据 McCabe 度量法，图 4-20 所示程序图的复杂性度量值为______。

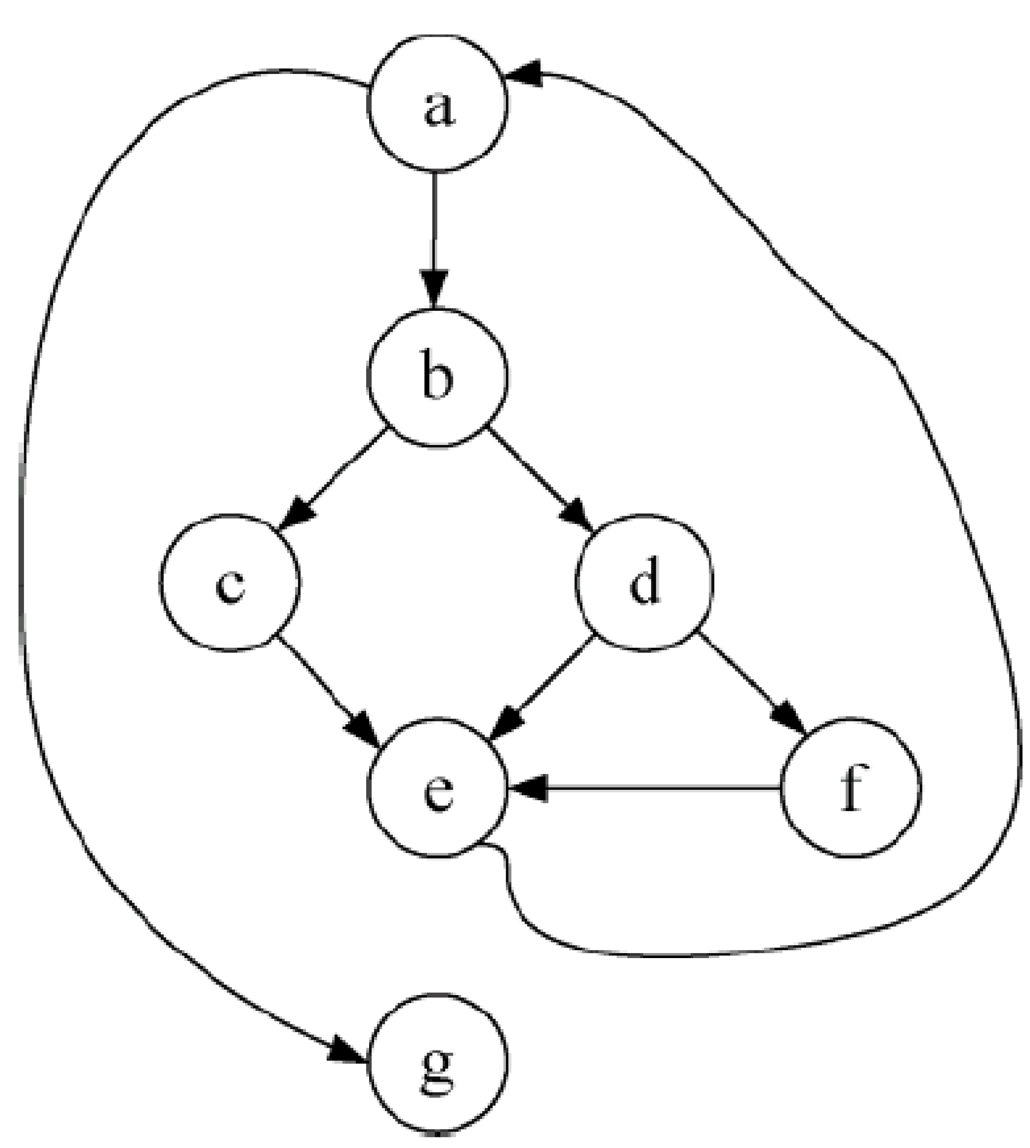


图 4-20　程序图的复杂性

A. 4　　B. 5　　C. 6　　D. 7

22. 在软件评审中，设计质量是指设计的规格说明书符合用户的要求。设计质量的评审内容不包括______。

A. 软件可靠性　　　　B. 软件的可测试性

C. 软件性能实现情况　　　　D. 模块层次

23. 冗余技术通常分为四类，其中______按照工作方法可以分为静态、动态和混合冗余。

A. 时间冗余　　B. 信息冗余　　C. 结构冗余　　D. 冗余附加技术

第 5 章

系统开发与运行

2014 年的《软件设计师教程(第 4 版)》将“软件开发与运行”拆分为两章：软件工程基础知识和系统开发与运行。

本章要点

- 需求分析
- 系统分析的主要步骤
- 结构化分析方法
- 概要设计与详细设计的基本任务
- 系统模块结构设计
- 系统测试
- 系统维护的类型

5.1 系统分析与设计概述

考核说明：本节主要介绍系统分析、系统设计、系统总体结构设计和系统软件。系统设计中的模块结构、系统文档是重点内容，考试中经常出现。

5.1.1 系统分析概述

1. 系统分析的目的和任务

系统分析的主要任务是对现行系统进一步调查，将调查中所得到的文档资料集中，对组织内部整体管理状况和信息处理过程进行分析，为系统开发提供所需资料，并提交系统方案说明书。最后，提出信息系统的各种设想和方案，并对所有的设想和方案进行分析、研究、比较、判断和选择，获得一个最优的新系统的逻辑模型，并在用户理解计算机系统的工作流程和处理方式的情况下，将它明确地表达成书面资料——系统分析报告，即系统方案说明书。

2. 系统分析的步骤

系统分析的过程一般如下。

(1) 认识、理解当前的现实环境，获得当前系统的“物理模型”。

(2) 从当前系统的“物理模型”抽象出当前系统的“逻辑模型”。

(3) 对当前系统的“逻辑模型”进行分析和优化，建立目标系统的“逻辑模型”。

(4) 对目标系统的逻辑模型具体化(物理化)，建立目标系统的物理模型。

系统分析阶段的主要工作步骤如下。

(1) 对当前系统进行详细调查，收集数据。

(2) 建立当前系统的逻辑模型。

(3) 对现状进行分析，提出改进意见和新系统应达到的目标。

(4) 建立新系统的逻辑模型。

(5) 编写系统方案说明书。

5.1.2 系统设计的基本原理

1. 抽象

抽象是一种设计技术，重点说明一个实体的本质方面，而忽略或者掩盖不很重要或非本质的方面。在进行模块化设计时也可以有多个抽象层次，最高抽象层次的模块用概括的方式叙述问题的解法，较低抽象层次的模块是对较高抽象层次模块关于问题解法描述的细化。

2. 模块化

模块化是指将一个待开发的软件分解成若干个小的简单部分——模块，每个模块可独

立地开发、测试，最后组装成完整的程序。模块化的目的是使程序的结构清晰，容易阅读、理解、测试和修改。

3. 信息隐蔽

信息隐蔽是开发整体程序结构时使用的法则，即将每个程序的成分隐蔽或封装在一个单一的设计模块中，以便定义每一个模块时尽可能少地显露其内部的处理。信息隐蔽原则对提高软件的可修改性、可测试性和可移植性都有重要的作用。

4. 模块独立

模块独立是指每个模块完成一个相对独立的特定子功能，并且与其他模块之间的联系简单。衡量模块独立程度的标准有两个：耦合和内聚。耦合是指模块之间联系的紧密程度。耦合度越高则模块的独立性越差。一般模块之间可能的耦合方式有 7 种，如图 5-1 所示。内聚是指模块内部各元素之间联系的紧密程度。内聚度越低，模块的独立性越差。一般模块的内聚性分为 7 种类型，如图 5-2 所示。因此，模块独立就是希望每个模块都是高内聚、低耦合的。



图 5-1　耦合的种类

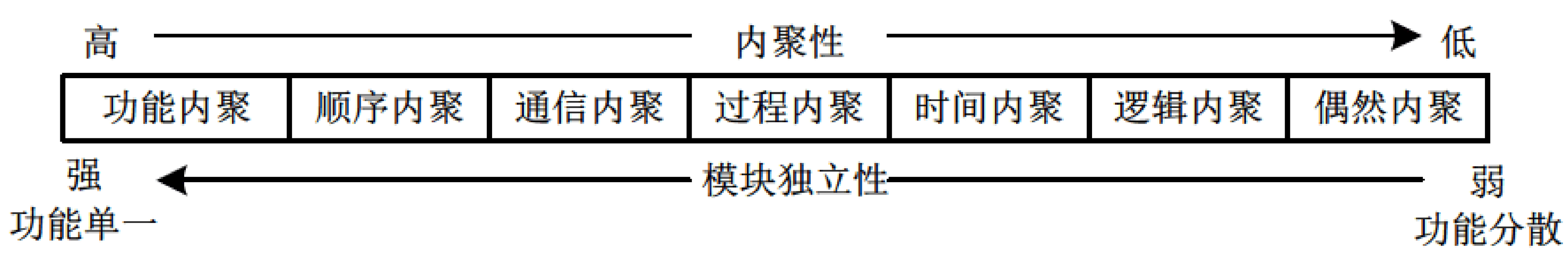


图 5-2　内聚的种类

真题链接

【例 5-1】模块 A、B 和 C 包含相同的 5 个语句，这些语句之间没有联系，为了避免重复，把这 5 个模块抽取出来组成模块 D。则模块 D 的内聚类型为__(16)__内聚。(2014 年 11 月真题 16)

(16)　A. 功能　　B. 通信　　C. 逻辑　　D. 巧合

【解析】功能内聚：完成一个单一功能，各个部分协同工作，缺一不可。

顺序内聚：处理元素相同，而且必须顺序执行。

通信内聚：所有处理元素集中在一个数据结构的区域上。

过程内聚：处理元素相关，而且必须按特定的次序执行。

时间内聚：所包含的任务必须在同一时间间隔内执行(如初始化模块)。

逻辑内聚：完成逻辑上相关的一组任务。

偶然内聚(巧合内聚)：完成一组没有关系或松散关系的任务。

【答案】D

【例 5-2】以下关于结构化开发方法的叙述中，不正确的是__(15)__。(2014 年 11 月真题 15)

(15) A. 总的指导思想是自顶向下，逐层分解
B. 基本原则是功能的分解与抽象
C. 与面向对象开发方法相比，更适合大规模、特别复杂的项目
D. 特别适合于数据处理领域的项目

【解析】结构化开发方法是一种传统开发方法，也是迄今为止信息系统中应用最普遍、最成熟的一种方法。其基本思想是：自顶向下、逐层分解。通常采用的模型是瀑布模型。其缺点是可重用性差、可维护性差，开发的软件往往不能满足用户的需要，需求分析处于被动状态，开发难度会因用户需求的变化而加大。面向对象开发方法的优点是与人类习惯的思维方法一致，稳定性好，可重用性好，可维护性好。因此，大型软件的开发一般都采用面向对象开发方法。

【答案】C

【例 5-3】模块 A 提供某个班级某门课程的成绩给模块 B，模块 B 计算平均成绩、最高分和最低分，将计算结果返回给模块 A，则模块 B 在软件结构图中属于<u>(16)</u>模块。(2014 年 5 月真题 16)

(16) A. 传入　　B. 传出　　C. 变换　　D. 协调

【解析】模块结构图中的模块类型如下。

(1)　传入模块：从下属模块取得数据，经过某些处理，再将其结果传给上级模块。

(2)　传出模块：从上级模块取得数据，经过某些处理，再将其结果传给下属模块。

(3)　变换模块：也叫加工模块。它从上级模块取得数据，进行特定的处理，转换成其他形式，再传回上级模块。它加工的数据流称作变换数据流。

(4)　协调模块：对所有下属模块进行协调和管理的模块。在系统的输入/输出部分或数据加工部分可以找到这样的模块。在一个好的模块结构图中，协调模块应在较高层出现。

【答案】C

5.1.3　系统设计的内容和步骤

系统设计的主要目的就是为系统制定蓝图，在各种技术和实施方法中权衡利弊，精心设计，合理使用各种资源，最终勾画出新系统的详细设计方案。

1. 概要设计的基本任务

1)　设计软件系统总体结构

设计软件系统总体结构的基本任务是：采用某种设计方法，将一个复杂的系统按功能划分成模块；确定每个模块的功能；确定模块之间的调用关系；确定模块之间的接口，即模块之间传递的信息；评价模块结构的质量。

2)　数据结构及数据库设计

(1)　数据结构的设计。在需求分析阶段，已经通过数据字典对数据的组成、操作约束和数据之间的关系等方面进行了描述，确定了数据的结构特性，在概要设计阶段要加以细化，详细设计阶段则规定具体的实现细节。在概要设计阶段，宜使用抽象的数据类型。

(2)　数据库的设计。数据库的设计是指数据存储文件的设计。

3)　编写概要设计文档

文档主要有概要设计说明书、数据库设计说明书、用户手册以及修订测试计划。

4)　评审

对设计部分是否完整地实现了需求中规定的功能、性能等要求，设计方法的可行性，关键的处理及内外部接口定义的正确性、有效性，各部分之间的一致性等都一一进行评审。

2. 详细设计的基本任务

详细设计的基本任务如下。

(1)　对每个模块进行详细的算法设计。用某种图形、表格和语言等工具将每个模块处理过程的详细算法描述出来。

(2)　对模块内的数据结构进行设计。

(3)　对数据库进行物理设计，即确定数据库的物理结构。

(4)　其他设计。根据软件系统的类型，还可能要进行以下设计。

- 代码设计。
- 输入输出格式设计。
- 用户界面设计。

(5)　编写详细设计说明书。

(6)　评审。对处理过程的算法和数据库的物理结构都要评审。

3. 处理过程设计

处理过程设计的关键是用一种合适的表示方法来描述每个模块的执行过程。这种表示方法应该简明、精确，并因此能直接导出用编程语言表示的程序。

1)　程序流程图

程序流程图包括三种基本成分：加工步骤，用方框表示；逻辑条件，用菱形表示；控制流，用箭头表示。

2)　盒图(NS 图)

在 NS 图中，每个处理步骤用一个盒子表示，盒子可以嵌套。盒子只能从上头进入，从下头走出，所以盒图限制了随意的控制转移，保证了程序的良好结构。

3)　形式语言

形式语言是用来描述模块具体算法的非正式的比较灵活的语言。形式语言的优点首先是接近自然语言，所以易于理解；其次，它可以作为注释嵌套在程序中成为内部文档，提高程序的自我描述性；最后，因为是语言形式，易于被计算机处理，可用行编辑程序或字处理系统对形式语言进行编辑修改。

4)　决策树

决策树是一种图形工具，适合于描述加工中具有多个策略而且每个策略和若干条件有关的逻辑功能。

5)　决策表

决策表是一种图形工具，呈表形。决策表可以将比较复杂的决策问题简洁地描述出来。

5.1.4 系统总体结构设计

1. 系统结构设计原则

系统结构设计应遵循如下原则。

- 分解-协调原则。
- 自顶向下原则。
- 信息隐蔽、抽象的原则。
- 一致性原则。
- 明确性原则。
- 模块之间的耦合尽可能小，模块的内聚度尽可能高。
- 模块的扇入系数和扇出系数要合理。
- 模块的规模适当。

2. 子系统划分

1) 划分原则

子系统划分要遵循如下原则。

- 子系统要具有相对独立性。
- 子系统之间数据的依赖性尽量小。
- 子系统划分的结果应使数据冗余较小。
- 子系统的设置应考虑今后管理发展的需要。
- 子系统的划分应便于系统分阶段实现。
- 子系统的划分应考虑到各类资源的充分利用。

2) 子系统结构设计

子系统结构设计的任务是确定划分后的子系统模块结构，并画出模块结构图。

3. 系统模块结构设计

1) 模块的概念

模块是组成系统的基本单位，它的特点是可以组合、分解和更换。系统中任何一个处理功能都可以看成一个模块。根据模块功能具体化程度的不同，可以分为逻辑模块和物理模块。

2) 模块结构图

模块结构图是结构化设计中描述系统结构的图形工具。作为一种文档，它必须严格地定义模块的名字、功能和接口，同时还应当在模块结构图上反映出结构化设计的思想。模块结构图由模块、调用、数据、控制和转接五种基本符号组成，如图 5-3 所示。

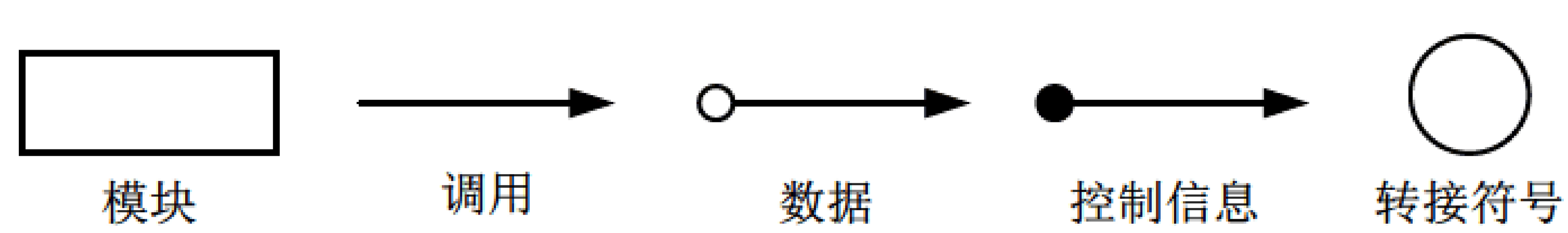


图 5-3 模块结构图的基本符号

如果一个模块是否调用一个从属模块，取决于调用模块内部的判断条件，则该调用模块间的判断调用采用菱形符号表示。如果一个模块通过其内部的循环功能来循环调用一个或多个从属模块，则该调用称为循环调用，用弧形箭头表示。

4. 数据存储设计

建立一个良好的数据组织结构和数据库，使整个系统都可以迅速、方便、准确地调用和管理所需的数据，是衡量信息系统开发工作好坏的主要指标之一。

数据结构组织和数据库或文件设计，就是要根据数据的不同用途、使用要求、统计渠道、安全保密性等，来决定数据的整体组织形式、表或文件的格式，以及决定数据的结构类别、载体、组织方式、保密级别等一系列问题。

5.1.5　系统文档

信息系统的文档不仅包括应用软件开发过程中产生的文档，还包括硬件采购和网络设计中形成的文档；不仅包括上述有一定格式要求的规范文档，也包括系统建设过程中的各种来往文件、会议纪要、会计单据等资料形成的不规范文档，后者是建设各方谈判甚至索赔的重要依据；不仅包括系统实施记录，也包括程序资料和培训教程等。

(1) 用户与系统分析人员在系统规划和系统分析阶段通过文档进行沟通。这里的文档主要包括可行性研究报告、总体规划报告、系统开发合同和系统方案说明书等。

(2) 系统开发人员与项目管理人员通过文档在项目期内进行沟通。这里的文档主要有系统开发计划、系统开发月报以及系统开发总结报告等项目管理文件。

(3) 系统测试人员与系统开发人员通过文档进行沟通。系统测试人员可以根据系统方案说明书、系统开发合同、系统设计说明书和测试计划等文档对系统开发人员所开发的系统进行测试。系统测试人员再将评估结果撰写成系统测试报告。

(4) 系统开发人员与用户在系统运行期间进行沟通。这里的文档主要是用户手册和操作指南。

(5) 系统开发人员与系统维护人员通过文档进行沟通。这里的文档主要有系统设计说明书和系统开发总结报告。

(6) 用户与维修人员在运行维护期间进行沟通。这里的文档主要有系统运行报告和维护修改建议、技术手册等。

真题链接

【例 5-4】以下关于文档的叙述中，不正确的是 (33) 。(2014 年 11 月真题 33)

(17) A. 项目相关人员可以通过文档进行沟通

B. 编写文档会降低软件开发的效率

C. 编写高质量文档可以提高软件开发的质量

D. 文档是软件的不可或缺的部分

【解析】在软件的不同阶段，需要编写不同的文档。因为我们在实际开发过程中，会发现之前设计不合理的地方或者是考虑不周的地方，这就需要对之前的文档进行修改。

【答案】B

【例 5-5】以下关于文档的叙述中，不正确的是 (33) 。(2014 年 5 月真题 33)

(33) A. 文档仅仅描述和规定了软件的使用范围及相关的操作命令

B. 文档也是软件产品的一部分，没有文档的软件就不能称为软件

C. 软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量

D. 高质量文档对于发挥软件产品的效益有着重要的意义

【解析】文档是软件开发使用和维护中的必备资料。文档能提高软件开发的效率，保证软件的质量，而且在软件的使用过程中有指导、帮助、解惑的作用，尤其在维护工作中，文档是不可或缺的资料。文档不仅包括软件开发过程中产生的文档，还包括硬件采购和网络设计中形成的文档；不仅包括有一定格式要求的规范文档，也包括系统建设过程中的各种往来文件、会议纪要、会计单据等资料形成的不规范文档。

【答案】A

5.2　需求分析与需求工程

考核说明：本节主要介绍需求分析和需求工程，考试中很少出现，偶尔在客观题中会考到需求分析。

1. 软件需求

软件需求是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。通常，这些需求包括功能需求、性能需求、用户或人的因素、环境需求、界面需求、文档需求、数据需求、资源使用需求、安全保密需求、可靠性需求、软件成本消耗与开发进度需求等，并预先估计以后系统可能达到的目标。

2. 需求分析原则

(1) 必须能够表示和理解问题的信息域。

(2) 必须能够定义软件将完成的任务。

(3) 必须能够表示软件的行为(作为外部事件的结束)。

(4) 必须划分描述数据、功能和行为的模型，从而可以分层次地揭示细节。

(5) 分析过程应该从要素信息移向细节信息。

3. 需求工程

需求工程是一个不断反复的需求定义、文档记录、需求演进的过程，并最终在验证的基础上冻结需求。需求工程可以细分为需求获取、需求分析与协商、系统建模、需求规约、需求验证以及需求管理 6 个阶段。

4. 需求建模

常用的分析和建模方法有面向数据流方法、面向数据结构方法和面向对象方法。

5. 需求规约与验证

软件需求规约是需求分析的最终产物，通常包含引言、信息描述、功能描述、行为描述、检验标准、参考书目、附录。

需求验证的目的是要检验需求是否能够反映用户的意愿。需求验证需要对需求文档中定义的需求执行多种检查。开发团队要对用户需求进行“遍访”，逐条解释需求含义；评审团队应该检查需求的有效性、一致性和作为一个整体的完备性。为了保证软件需求定义的质量，验证应该由专门的人员来负责，按照规定严格进行。除分析人员之外，还要有用户，开发部门的管理者，软件设计、实现、测试的人员参加。

6. 需求管理

需求管理是对需求工程所有相关活动的规划和控制。在需求管理中，每个需求被赋予唯一的标识符，一旦标识出需求，就可以为需求建立跟踪表，每个跟踪表标识需求与其他需求或设计文档、代码、测试用例的不同版本间的关系。

真题链接

【例 5-6】“软件产品必须能够在 3 秒内对用户请求做出响应”属于软件需求中的 (18) 。(2013 年 5 月真题 18)

(18) A. 功能需求　　B. 非功能需求　　C. 设计约束　　D. 逻辑需求

【解析】软件需求中对软件产品的响应时间、吞吐量、价格等属性的要求都属于非功能性需求。

【答案】B

5.3　结构化分析方法

考核说明：本节主要介绍结构化分析方法。数据流图是下午科目必考内容，难度不大，重在理解。

结构化分析(Structure Analysis，SA)方法是一种面向数据流的需求分析方法，适用于分析大型数据处理系统，是一种简单、实用的方法，现在已经得到广泛的应用。

结构化分析方法的基本思想是自顶向下逐层分解。SA 方法的分析结果由以下几部分组成：一套分层的数据流图、一本数据字典、一组说明和补充材料。

1. 基本概念

1)　数据流图

数据流图或称数据流程图(DFD)是一种便于用户理解、分析系统数据流程的图形工具。它摆脱了系统的物理内容，可以精确地在逻辑上描述系统的功能、输入、输出和数据存储等，是系统逻辑模型的重要组成部分。

2)　数据字典

数据流图描述了系统的分解，但没有对图中各成分进行说明。数据字典就是为数据流图中的每个数据流、文件、加工，以及组成数据流或文件的数据项做出说明。

数据字典的内容：数据流条目、文件条目和数据项条目。

数据字典管理：主要是把字典条目按照某种格式组织后存储在字典中，并提供排序、查找、统计等功能。

3) 加工逻辑的描述

常用的加工逻辑描述方法有结构化语言、判定表和判定树三种。其中结构化语言是一种介于自然语言和形式化语言之间的半形式化语言，是自然语言的一个受限子集。结构化语言没有严格的语法，它的结构通常可分为内外两层。外层有严格的语法，内层的语法比较灵活，可以接近自然语言的描述。

2. DFD 的基本成分

DFD 的基本成分及其图形表示方法如图 5-4 所示。

(1) 外部实体。外部实体是指存在于软件系统之外的人员或组织，它指出了系统所需数据的发源地和系统所产生的数据的归宿地。

(2) 加工。加工描述了输入数据流到输出数据流之间的变换，也就是输入数据流经过什么处理后变成了输出数据流。每个加工都有一个名字和编号。编号能反映出该加工位于分层 DFD 中的哪个层次和哪张图中，也能够看出它是哪个加工分解出来的子加工。

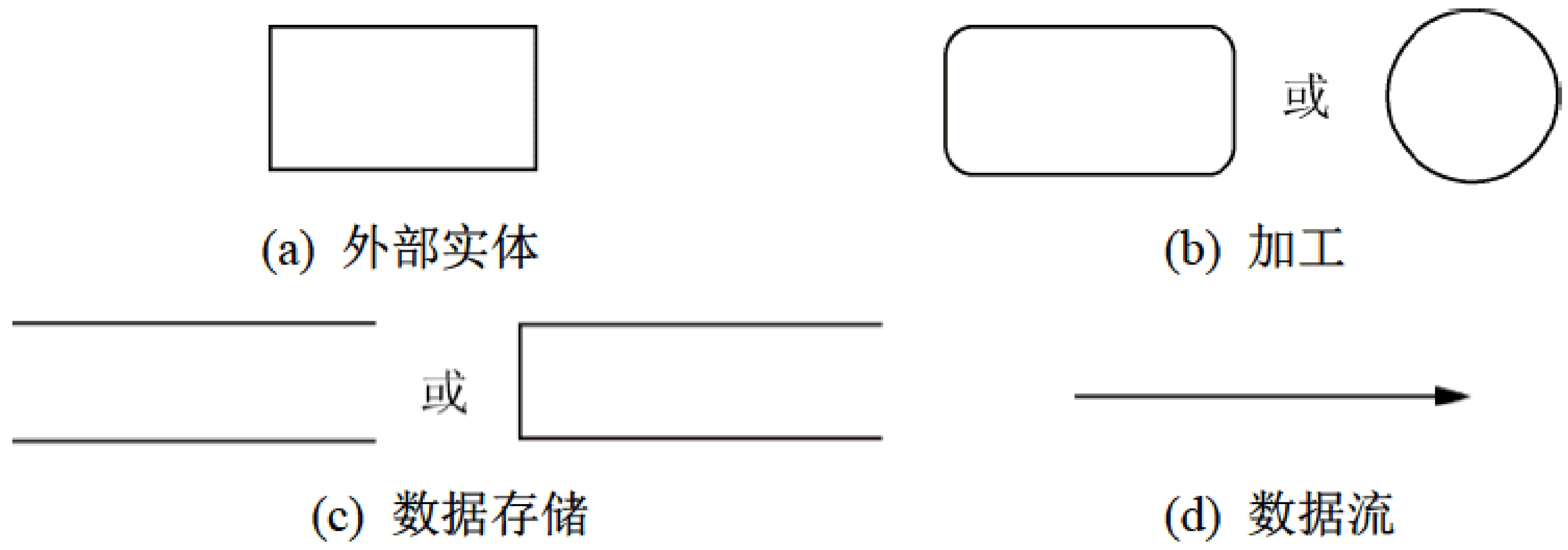


图 5-4 DFD 的基本成分

(3) 数据存储。数据存储用来表示存储的数据，每个数据存储都有一个名字。DFD 中的数据存储在具体实现时可以用文件系统实现，也可以用数据库系统实现。数据存储的存储介质可以是磁盘、磁带或其他存储介质。

(4) 数据流。数据流由一组固定成分的数据组成，表示数据的流向。值得注意的是，DFD 中描述的是数据流，而不是控制流。除了流向数据存储或从数据存储流出的数据流不必命名外，每个数据流都必须有一个合适的名字，以反映该数据流的含义。

3. 数据流图的扩充符号

星号(*)表示数据流之间存在“与”关系。

加号(+)表示数据流之间存在“或”关系。

异或(⊕)表示数据流之间存在“互斥”关系。

4. 分层数据流图的画法

下面介绍分层数据流图的画法。

(1) 画系统的输入和输出。把整个软件系统看作一个大的加工，然后根据系统从哪些

外部实体接收数据流，以及系统发送数据流到哪些外部实体，就可以画出系统的输入和输出图，这张图称为顶层图。

(2) 画系统的内部。将顶层图的加工分解成若干个加工，并用数据流将这些加工连接起来，使得顶层图中的输入数据经过若干个加工处理后变换成顶层图的输出数据流。这张图称为 0 层图。从一个加工画出一张数据流图的过程实际上就是对这个加工的分解。

可以用下述方法来确定加工：在数据流的组成或值发生变化的地方画一个加工，这个加工的功能就是实现这一变化；也可根据系统的功能确定加工。

确定数据流的方法：当用户把若干个数据看作一个单位来处理(这些数据一起到达，一起加工)时，可把这些数据看成一个数据流。

对于一些以后某个时间要使用的数据可以组织成一个数据存储来表示。

(3) 画加工的内部。把每个加工看作一个小系统，该加工的输入/输出数据流看成小系统的输入/输出数据流，于是可以用与画 0 层图同样的方法画出每个加工的 DFD 子图。

对第(3)步分解出来的 DFD 子图中的每个加工，重复第(3)步的分解，直至图中尚未分解的加工都足够简单(也就是说这种加工不必再分解)为止。至此，就得到了一套分层数据流图。

5. 对图和加工进行编号

对于一个软件系统，其数据流图可能有许多层，每一层又有许多张图。为了区分不同的加工和不同的 DFD 子图，应该对每张图和每个加工进行编号，以利于管理。

1) 父图与子图

假设分层数据流图中的某张图(记为图 A)中的某个加工可用另一张图(记为图 B)来分解，则称图 A 是图 B 的父图，图 B 是图 A 的子图。在一张图中，有些加工需要进一步分解，有些加工则不必分解。因此，如果父图中有 n 个加工，那么它可以有 0～n 张子图(这些子图位于同一层)，但每张子图都只对应于一张父图。

2) 编号

顶层图只有一张，图中的加工也只有一个，所以不必编号。

0 层图只有一张，图中的加工号可以分别是 0.1、0.2…或者是 1、2…

子图号就是父图中被分解的加工号。

图的加工号由图号、圆点和序号组成。

6. 应注意的问题

(1) 适当地为数据流、加工、数据存储、外部实体命名，名字应反映该成分的实际含义，避免空洞的名字。

(2) 画数据流而不要画控制流。

(3) 每条数据流的输入或者输出都是加工。

(4) 一个加工的输出数据流不应与输入数据流同名，即使它们的组成成分相同。

(5) 允许一个加工有多条数据流流向另一个加工，也允许一个加工有两个相同的输出数据流流向两个不同的加工。

(6) 保持父图与子图平衡。也就是说，父图中某加工的输入/输出数据流必须与它的子图的输入/输出数据流在数量和名字上相同。值得注意的是，如果父图的一个输入(或输出)

数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一个数据流，那么它们仍然算是平衡的。

(7) 在自顶向下的分解过程中，若一个数据存储首次出现时只与一个加工有关，那么这个数据存储应作为这个加工的内部文件而不必画出。

(8) 保持数据守恒。也就是说，一个加工所有输出数据流中的数据必须能从该加工的输入数据流中直接获得，或者是通过该加工产生的数据。

(9) 每个加工必须既有输入数据流，又有输出数据流。

(10) 在整套数据流图中，每个数据存储必须既有读的数据流，又有写的数据流。但在某一张子图中可能是只有读没有写，或者是只有写没有读。

7. 补充和完善数据流

补充和完善数据流是数据流图最常出的题型，也是重点和难点。解答此类问题有一定的技巧，以一些常规的入口作为突破口，往往能事半功倍。

遇到这类问题，首先要想到分层数据流图的数据流平衡原则，即父图和子图的输入/输出数据流一致，这是找出遗漏数据流非常重要的技巧。其次，每个加工至少有一个输入流和一个输出流，反映此加工的数据来源和结果，加工的输出数据流应该都有其对应的输入数据流。最后，要找出遗漏的数据流，最根本的依据还是说明。因为除了图之外，题目中最重要的部分就是说明，因为说明部分详细介绍了系统的功能，所以是找出所缺数据流的基本入口。

有时数据流平衡原则不作为解题的直接方法，而作为排除的手段，然后根据说明或其他方法找到图中遗漏的数据流。

8. 找出错误或多余的数据流

要找出错误或多余的数据流，解题方法可以参考完善数据流的方法。一般可以先进行上下层图的对照和分析，然后检查是否每个加工至少有一个输入流和一个输出流，是否加工的输出数据流都有其对应的输入数据流。而最根本的判断标准仍然是题目的说明部分。所以考生一定要耐心、认真地阅读题目中对系统功能的阐述和说明，然后解题时再次阅读说明，从中找到依据和突破口。

9. 找出多余的文件

在某层数据流图中，只画流程图各加工之间的公共数据文件时，如果一个文件仅仅作用于一个加工，即和该文件有关的输入/输出数据流只涉及一个加工，那么该文件可以作为局部文件出现在该加工的子图中，在父图中则可以省略。这个规则是为了使整个流程图的层次结构更为清晰、科学。当然这些文件如果画出，并不会造成理解错误。

另外，如果某层图只有一层细化图，即该层图没有子图，则不存在局部文件和外部文件之分，其中涉及的任何文件都不作为多余的文件。

10. 添加数据字典条目

此类题一般难度比较小，可以根据说明部分找出答案。同时还可以结合给出的数据流图，查看有关记录需要输入给哪些加工，这些加工输出哪些字段。

11. 系统分析报告

在系统分析报告中，数据流图、数据字典和加工说明这三个部分是主体，是系统分析报告中必不可少的组成部分。系统分析报告主要有以下三个作用。

(1) 描述了目标系统的逻辑模型，作为开发人员进行系统设计和实施的基础。

(2) 作为用户和开发人员之间的协议或合同，为双方的交流和监督提供基础。

(3) 作为目标系统验收和评价的依据。

一个完整的系统分析报告包括下述内容。

(1) 组织情况概述。

(2) 现行系统概述。

(3) 系统逻辑模型。

(4) 新系统在各个业务处理环节拟采用的管理方法、算法或模型。

(5) 与新的系统相配套的管理制度和运行机制的建立。

(6) 系统设计与实施的初步计划。

(7) 用户方负责人审批意见。

真题链接

【例 5-7】以下关于数据流图中基本加工的叙述，不正确的是__(15)__。(2013 年 11 月真题 15)

(15) A. 对每一个基本加工，必须有一个加工规格说明

B. 加工规格说明必须描述把输入数据流变换为输出数据流的加工规则

C. 加工规格说明必须描述实现加工的具体流程

D. 决策表可以用来表示加工规格说明

【解析】数据流图中的守恒加工原则：

① 对同一个加工来说，输入与输出的名字必须不相同，即使它们的组成成分相同。

② 对于每个加工，必须既有输入数据流，又有输出数据流。

加工规格说明的主要目的是要表达“做什么”，而不是“怎么做”，所以本题答案选 C。

【答案】C

【例 5-8】阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。(2014 年 11 月下午试题一)

【说明】

某大型披萨加工和销售商为了有效管理生产和销售情况，欲开发一披萨信息系统，其主要功能如下：

(1) 销售。处理客户的订单信息，生成销售订单，并将其记录在销售订单表中。销售订单记录了订购者、所订购的披萨、期望的交付日期等信息。

(2) 生产控制。根据销售订单以及库存的披萨数量，制定披萨生产计划(包括生产哪些披萨、生产顺序和生产量等)，并将其保存在生产计划表中。

(3) 生产。根据生产计划和配方表中的披萨配方，向库存发出原材料申领单，将制作好的披萨的信息存入库存表中，以便及时进行交付。

(4) 采购。根据所需原材料及库存量，确定采购数量，向供应商发送采购订单，并将其记录在采购订单表中；得到供应商的供应量，将原材料数量记录在库存表中，在采购订单表中标记已完成采购的订单。

(5) 运送。根据销售订单将披萨交付给客户，并记录在交付记录表中。

(6) 财务管理。在披萨交付后，为客户开具费用清单，收款并出具收据：依据完成的采购订单给供应商支付原材料费用并出具支付细节；将收款和支付记录存入收支记录表中。

(7) 存储。检查库存的原材料、披萨和未完成订单，确定所需原材料。

现采用结构化方法对披萨信息系统进行分析与设计，获得如图 1-1 所示的上下文数据流图和图 1-2 所示的 0 层数据流图。

【问题 1】

根据说明中的词语，给出图 5-5 中的实体 E1～E2 的名称。

【问题 2】.

根据说明中的词语，给出图 5-6 中的数据存储 D1～D5 的名称。

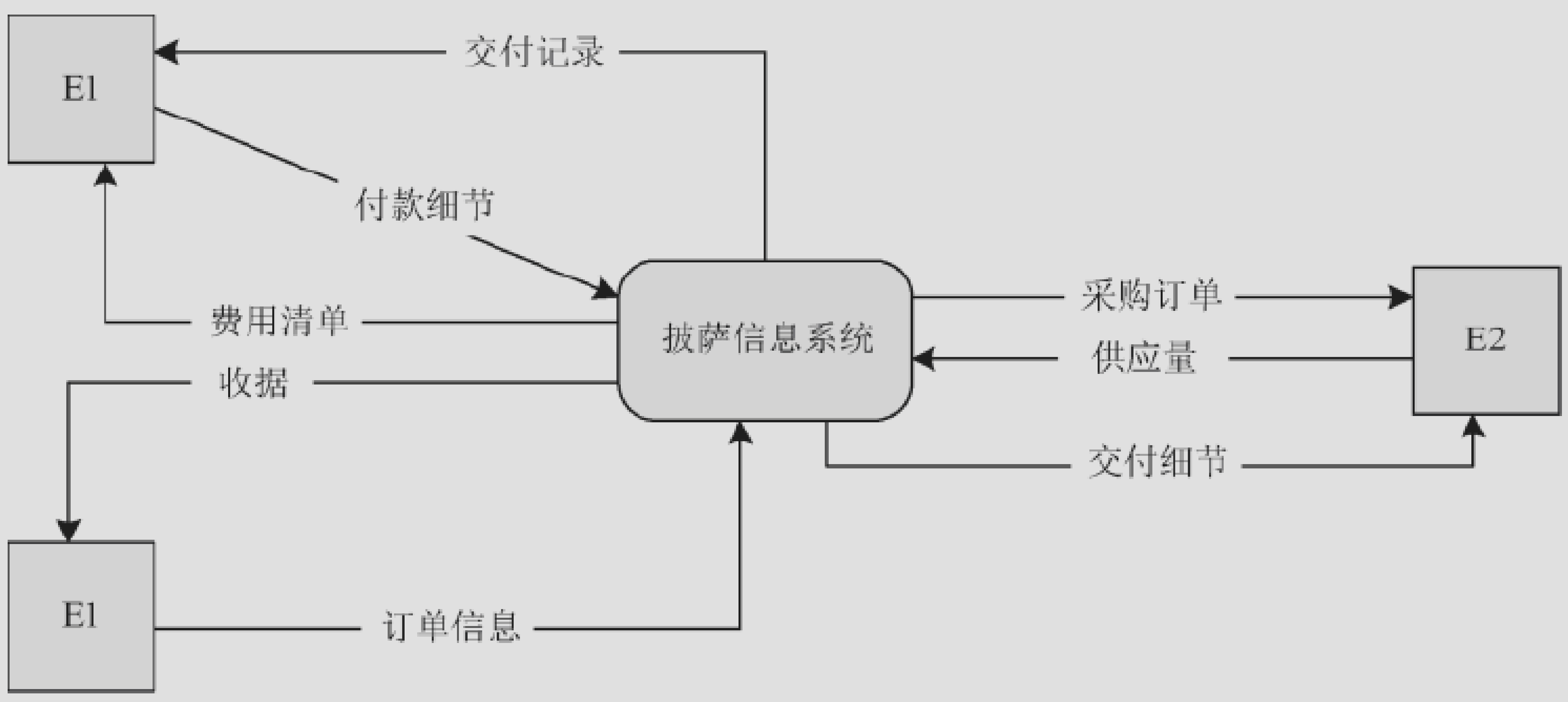


图 5-5　上下文数据流图

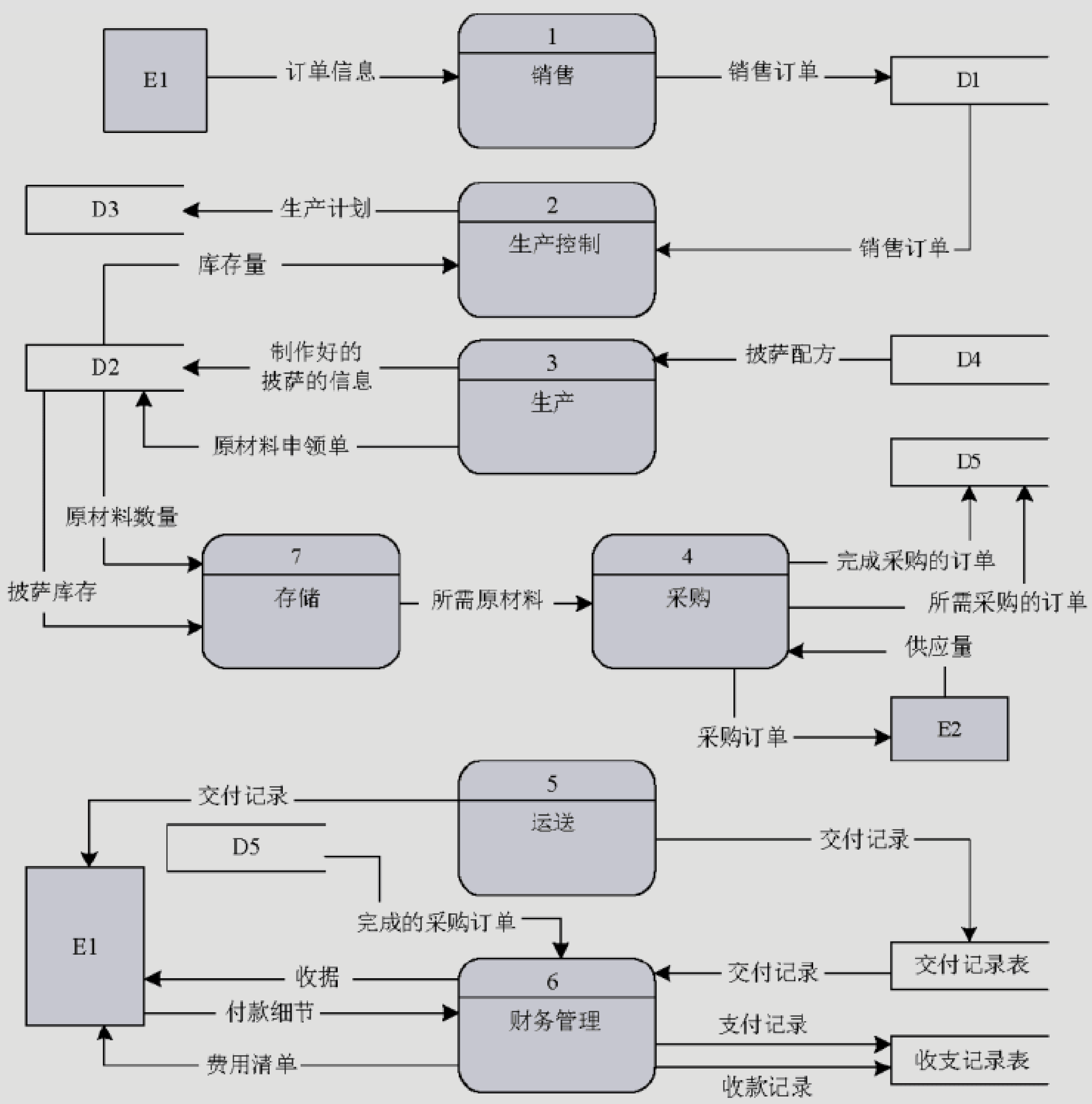


图 5-6　0 层数据流图

【问题 3】

根据说明中的词语，补充图 5-6 中缺失的数据流及其起点和终点。

【解析】

该题以披萨信息系统为载体来考查学生对数据流图知识点的掌握程度。从题目的问答形式上来看，和往年相似，要求补充外部实体、补充缺失数据流、找出外部存储。

【问题 1】

根据 0 层数据流中财务管理为客户开具费用清单数据流可知，E1 实体为客户；从向供应商发送采购订单、得到供应商的供应量可知，E2 实体为供应商。

【问题 2】

根据题中对系统功能的描述，生产控制中要制定生产计划，并将其保存在生产计划表中可知，0 层数据流图中的数据流“生产计划”的终点 D3 为生产计划表；生产中要将制作好的披萨的信息存入库存表中，可知 D2 为库存表；采购时，在采购订单表中标记已完成的订单，可知 D5 为采购订单表；“生产功能中”，由“根据生产计划和配方表中的披萨配方，向库存发出原料申请”可知，D4 为配方表；“销售”功能中，根据“处理客户的订单信息，生成销售订单。并将其记录在销售订单表中”可知，D1 为销售订单表。

【问题 3】

本题考查补充分层 DFD 中的数据流。在分层 DFD 中，需要保持父图和子图的平衡，即父图中某加工的输入输出数据流必须与其子图的输入输出数据流在数量和名字上相同，或者父图的一个输入(或输出)数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是附图中的一个数据流。由“财务管理”段中“依据完成的采购订单给供应商支付原材料费用并出具支付细节”的描述，存在一个起点为财务管理，重点为供应商即 E2 的数据流，即支付细节数据流；由“运送”段中“根据销售订单将披萨交付给客户，并记录在交付记录表中”可知存在一个由“销售订单”指向“运送”的数据流，即销售订单数据流；由“采购”段中“根据所需原材料及库存量，确定采购数量”可知存在由“库存表”指向“采购”的数据流，数据流为库存量；根据“存储”段中“检查库存的原材料、披萨和未完成订单，确定所需原料”可知，存在一个由“订单表”指向“存储”的数据流，数据流为未完成的订单。

【答案】

【问题 1】

E1：客户；E2：供应商

【问题 2】

D1：销售订单表；D2：库存表；D3：生产计划表；D4：配方表；D5：采购订单表

【问题 3】

(1) 数据流名称：支付细节；起点：财务管理；终点：E2

(2) 数据流名称：销售订单；起点：销售订单表；终点：5 运送

(3) 数据流名称：生产计划；起点：D3；终点：3 生产

(4) 数据流名称：库存量；起点：D2；终点：4 采购

(5) 数据流名称：原材料数量；起点：4 采购；终点：D2

(6) 数据流名称：未完成订单；起点：销售订单表；终点：7 存储。

【例 5-9】阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。(2014 年 5 月下午试题一)

【说明】

某巴士维修连锁公司欲开发巴士维修系统，以维护与维修相关的信息，该系统的主要功能如下：

(1) 记录巴士 ID 和维修问题。巴士到车库进行维修，系统将巴士基本信息和 ID 记录在巴士列表文件

中，将待维修机械问题记录在维修记录文件中，并生成维修订单。

(2) 确定所需部件，根据维修订单确定维修所需部件，并在部件清单中进行标记。

(3) 完成维修，机械师根据维修记录文件中的待维修机械问题，完成对巴士的维修，登记维修情况；将机械问题维修情况记录在维修记录文件中，将所用部件记录在部件清单中，并将所用部件清单发送给库存管理系统以对部件的使用情况进行监控，巴士司机可查看已维修机械问题。

(4) 记录维修工时，将机械师提供的维修工时记录在人事档案中，将维修总结发送给主管进行绩效考核。

(5) 计算维修总成本，计算部件清单中实际所用部件、人事档案中所用维修工时的总成本；将维修工时和所用部件成本详细信息给会计进行计费。

现采用结构化方法对巴士维修系统进行分析与设计，获得如图 5-7 所示的上下文数据流图和图 5-8 所示的 0 层数据流图。

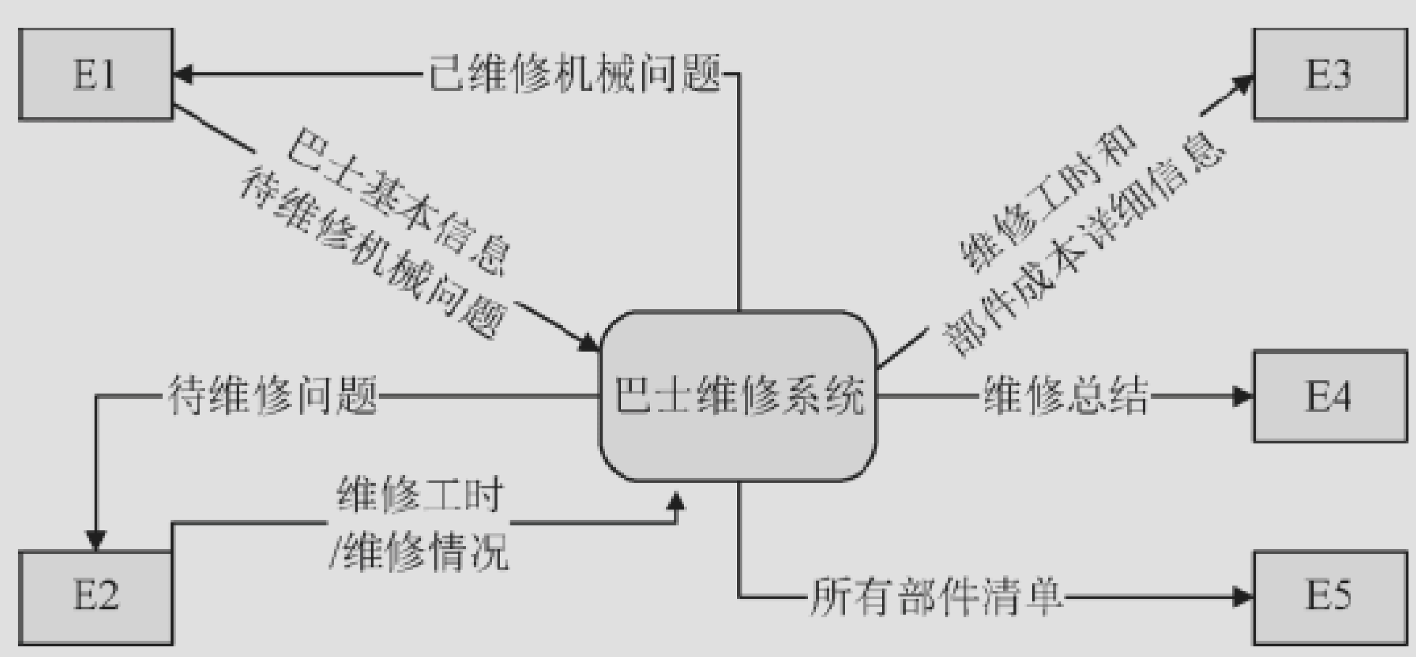


图 5-7　上下文数据流图

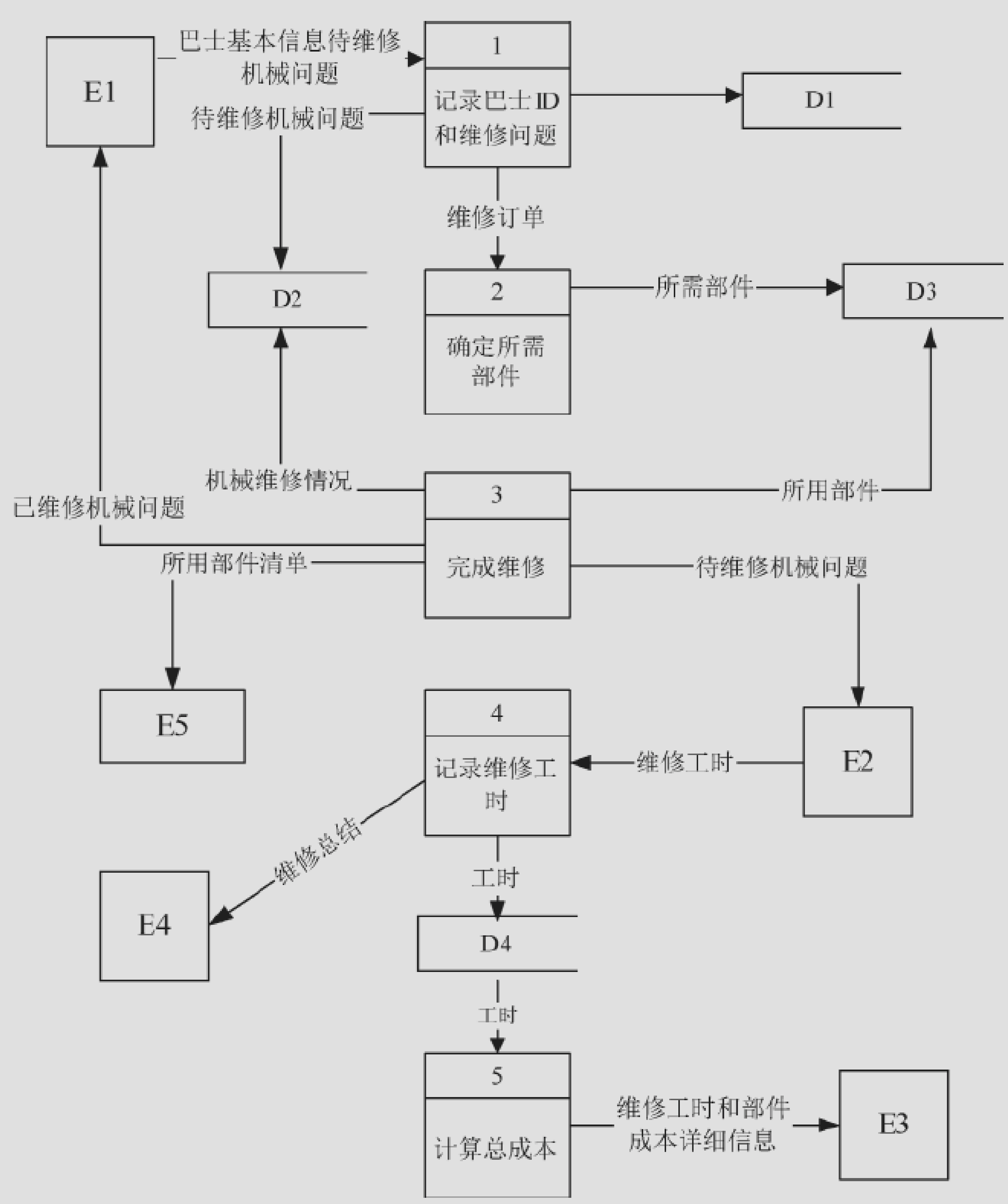


图 5-8　0 层数据流图

【问题 1】

使用说明中的词语，给出图 5-7 中的实体 E1～E5 的名称。

【问题 2】

使用说明中的词语，给出图 5-8 中的数据存储 D1～D4 的名称。

【问题 3】

说明图 5-8 中所存在的问题。

【问题 4】

根据说明和图中术语，采用补充数据流的方式，改正图 5-8 中的问题。要求给出所补充数据流的名称、起点和终点。

【解析】本题考查数据流图(DFD)的应用，采用结构化方法进行系统分析与设计，是一道传统题目，要求考生细心分析题目中所描述的内容。DFD 是一种便于用户理解、分析系统数据流程的图形化建模工具，是系统逻辑模型的重要组成部分。

【问题 1】本问题考查顶层 DFD。顶层 DFD 一般用来确定系统边界。结合图 5-7 上下文数据流图以及根据系统功能的描述(3)“巴士司机可查看已维修问题”可知，E1 为“巴士司机”；根据系统功能描述(4)“机械师提供的维修工时记录在人事档案中”和“维修总结发送给主管进行考核”可知，E2 为“机械师”，E4 为“主管”；根据系统功能描述(5)“维修工时和所用成本详细信息给会计进行计费”可知，E3 为“会计师”，由“所用部件清单发送给库存管理系统”可推断，E5 为库存管理系统。

【问题 2】本问题考查 DFD 中数据存储的确定。本题涉及的数据存储有 5 个，即巴士列表文件、维修记录文件、部件清单和人事档案。接下来需要结合图 5-8 所示的 0 层数据流进行确定。D1 与“记录巴士 ID 和维修问题”有关，根据题意，D1 是“巴士列表文件”；待维修机械问题记录在 D2 中，可知，D2 为“维修记录文件”；所需部件记录在 D3 中，由题意可知 D3 为“部件清单”；D4 记录的是工时，由题意，D4 是“人事档案”。

【问题 3】本问题考查 DFD 中的数据流。在分查 DFD 中，需要保持父图和子图的平衡，即父图中某加工的输入/输出数据流必须与其子图的输入/输出数据流在数量和名字上相同，或者父图的一个输入(或输出)数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是父图中的一个数据流。由(5)的描述“计算部件清单中实际所用部件、人事档案中所用维修工时的总成本”可知，缺少关于实际所用部件数据流，因此增加一个数据流实际所用部件，起点：D3，终点：5 计算总成本。根据“完成维修”功能中的描述，机械师可根据维修记录文件中的待维修机械问题，完成巴士维修，因此缺少数据流“待维修机械问题”，起点：D2，终点：3 完成维修。

【问题 4】根据问题 3 解析，补充以下数据流：实际所用部件，起点：D3，终点：5 计算总成本；待维修机械问题”，起点：D2，终点：3 完成维修。

【答案】

【问题 1】

E1：巴士司机

E2：机械师

E3：会计

E4：主管

E5：库存管理系统

【问题 2】

D1：巴士列表文件

D2：维修记录文件

D3：部件清单

D4：人事档案

【问题 3】

缺失以下数据流：

(1) 名称：待维修机械问题；起点：D2；终点：3 完成维修。

(2) 名称：实际所用部件；起点：D3；终点：5 计算总成本。

【问题 4】

补充以下数据流：

(1) 名称：待维修机械问题；起点：D2；终点：3 完成维修。

(2) 名称：实际所用部件；起点：D3；终点：5 计算总成本。

5.4 结构化设计方法

考核说明：本节主要介绍结构化设计的步骤和数据流图到软件体系结构的映射，信息流和变换分析是难点，需要重点理解和掌握。

结构化设计(SD)方法是一种面向数据流的设计方法，它可以与 SA 方法链接。结构化设计方法的基本思想是将系统设计成由相对独立、功能单一的模块组成的结构。结构化设计方法中用结构图(Structure Chart)来描述软件系统的体系结构，并指出一个软件系统由哪些模块组成，以及模块之间的调用关系。

5.4.1 结构化设计的步骤

(1) 建立初始结构图。通过功能分解将整个软件分解成若干个较小的功能模块，每个较小的功能模块还可以进一步分解，直到得到一组不必再分解的模块。一个模块的大小以 50～100 行程序代码为宜。

(2) 对结构图的改进。

(3) 书写设计文档。

(4) 设计评审。

5.4.2 数据流图到软件体系结构的映射

1. 信息流的类型

在需求分析阶段，用 SA 方法产生了数据流图。面向数据流的设计能方便地将 DFD 转换成程序结构图。DFD 中从系统的输入数据流到系统的输出数据流的一连串连续变换形成

了一条信息流。DFD 的信息流大体上可以分为两种类型，一种是变换流，另一种是事务流。

2. 变换分析

变换分析是从变换流型的 DFD 导出程序结构图。

(1) 确定输入流和输出流，孤立出变换中心。DFD 中从物理输入到逻辑输入的部分构成系统的输入流，从逻辑输出到物理输出的部分构成系统的输出流，位于输入流和输出流之间的部分就是变换中心。

(2) 第一级分解：主要是设计模块结构的顶层和第一层。

(3) 第二级分解：主要是设计中、下层模块。

(4) 事务分析：是从事务型 DFD 导出程序图。

(5) SD 方法的设计步骤：①复查并精化数据流图；②确定 DFD 的信息流类型(变换流或事务流)；③根据流类型分别实施变换分析或事务分析；④根据系统设计的原则对程序结构图进行优化。

5.5 Web 应用系统分析与设计

考核说明：本节内容是第 4 版教程中增加的部分。之前考试没有出现相关的考题。

5.5.1 WebApp 的特性

绝大多数 WebApp 具备下列属性。

(1) 网络密集性。WebApp 驻留在网络上，服务于不同客户全体的需求。

(2) 并发性。大量用户可能同时访问 WebApp。

(3) 无法预知的负载量。WebApp 的用户数量每天都可能有数量级的变化。

(4) 性能。如果一位 WebApp 用户必须等待很长时间，该用户就可能转向其他地方。

(5) 可用性。对于热门的 WebApp，用户通常要求能够 24/7/365(全天候)访问。

(6) 数据驱动。许多 WebApp 的主要功能是使用超媒体向最终用户提供文本、图片、音频及视频内容。

5.5.2 Web 应用系统分析模型

1. 内容模型

内容模型给出由 Web 应用系统提供的全部系列内容，包括文字、图形、图像、音频和视频。内容的开发可能发生在 Web 应用系统实现之前、Web 应用系统构建之中，或者 Web 应用系统投入运行以后的很长一段时间里。内容模型必须具备描述内容对象构件的能力。数据树表述了常用于描述某个构件的一种信息层次，不带阴影的长方形表示简单或复合数据项(一个或多个数据值)，带阴影的长方形表示内容对象。

2. 交互模型

交互模型描述了用户与 Web 应用系统采用了哪种交互方式。交互模型由一种或多种元素构成，包括用例、顺序图、状态图、用户界面原型等。

3. 功能模型

功能模型定义了将用于 Web 应用系统内容并描述其他处理功能的操作，这些处理功能不依赖于内容却是最终用户所必需的。功能模型描述 Web 应用系统的两个处理元素，每个处理元素代表抽象过程的不同层次：①用户可观察到的功能是由 Web 应用系统传递给最终用户的；②分析类中的操作实现与类相关的行为。

4. 导航模型

导航模型为 Web 应用系统定义所有导航策略。导航模型考虑了每一类用户如何从一个 Web 应用系统元素(例如内容对象)导航到另一个元素。

5. 配置模型

配置模型描述 Web 应用系统所在的环境和基础设施。在必须考虑复杂配置体系结构的情况下，可以使用 UML 部署图。

5.6 用户界面设计

考核说明：在第 4 版教程中对用户界面展开了介绍，之前的版本中涉及的内容不多，考试中基本没有考到。

1. 用户界面设计的黄金原则

用户界面设计是系统与用户之间的接口，也是控制和选择信息输入输出的主要途径。用户界面设计应坚持友好、简便、实用、易于操作的原则。

Theo Mandel 在其关于界面设计的著作中提出了 3 条“黄金原则”：用户操纵控制；减少用户的记忆负担；保持界面一致。

Mandel 定义了一组设计原则，允许用户操纵控制。

(1) 以不强迫用户进入不必要的或不希望的动作的方式来定义交互模式。

(2) 提供灵活的交互。

(3) 允许中断和撤销用户交互。

(4) 当技能级别增长时可以使交互流线化并允许定制交互。

(5) 使用户与内部技术细节隔离开来。

(6) 设计应允许用户与出现在屏幕上的对象直接交互。

Mandel 定义了一组设计原则，使得界面能够减少用户记忆负担。

(1) 减少对短期记忆的要求。

(2) 建立有意义的默认。

(3) 定义直观的快捷方法。

(4) 界面的视觉布局应该基于真实世界的象征。

(5) 以不断进展的方式揭示信息。

Mandel 定义了一组帮助保持界面一致性的设计原则。

(1) 允许用户将当前任务放入有意义的环境中。

(2) 在应用系统家族内保持一致性。

(3) 如果过去的交互模型已经建立起了用户期望，除非有不得已的理由，否则不要改变它。

2. 用户界面的分析与设计

用户界面分析和设计模型包括：

(1) 软件工程师所创建的设计模型(design model)。

(2) 人机界面设计工程师创建的用户模型(user model)。

(3) 最终用户在脑海里对界面产生的映像，称为用户的心理模型或系统感觉(system perception)。

(4) 系统实现者创建的系统映像(system image)。

用户界面的分析与设计过程是迭代的，包括 4 个不同的框架活动：界面分析及建模、界面设计、界面构造和界面确认。

3. 用户界面设计问题

在进行用户界面设计时，几乎总会遇到以下 4 个问题：系统响应时间、帮助设施、错误信息处理、菜单和命令标记。

5.7　测试基础知识

考核说明：本节主要介绍系统测试的目的、测试策略和测试方法、调试等内容，系统测试是必考内容，要重点掌握，尤其是白盒测试和黑盒测试。Web 应用系统测试是第 4 版教程中增加的内容，应有所了解。

5.7.1　系统测试与调试

1. 系统测试的意义和目的

系统测试是为了发现错误而执行程序的过程。或者说，软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例(即输入数据及其预期的输出结果)，并利用这些测试用例去运行程序，以发现程序错误的过程。

信息系统测试应包括软件测试、硬件测试和网络测试。系统测试是保证系统质量和可靠性的关键步骤，是对系统开发过程中的系统分析、系统设计和实施的最后复查。根据测试的概念和目的，在进行信息系统测试时应遵循以下基本原则。

- 应尽早并不断地进行测试。
- 测试工作应该避免由原开发软件的人或小组承担。

- 设计测试方案的时候，不仅要确定输入数据，而且要根据系统功能确定预期的输出结果。
- 在设计用例时，不仅要设计有效合理的输入条件，也要包含不合理、失效的输入条件。
- 在测试程序时，不仅要检验程序是否做了该做的事，还要检验程序是否做了不该做的事。
- 严格按照测试计划来进行，避免测试的随意性。
- 妥善保存测试计划、测试用例，作为软件文档的组成部分，为维护提供方便。
- 测试用例都是精心设计出来的，可以为重新测试或追加测试提供方便。

2. 测试过程

测试是开发过程中一个独立且非常重要的阶段，测试过程基本上与开发过程平行进行。一个规范的测试过程通常包括制订测试计划、编制测试大纲、根据测试大纲设计和生成测试用例、实施测试和生成测试报告。

真题链接

【例 5-10】在软件开发过程中，系统测试阶段的测试目标来自于<u>(32)</u>阶段。(2014 年 11 月真题 32)

(32) A. 需求分析　　B. 概要设计　　C. 详细设计　　D. 软件实现

【解析】系统测试是针对整个产品系统进行的测试，目的是验证系统是否满足了需求规格的定义，找出与需求规格不符或与之矛盾的地方，从而提出更加完善的方案。系统测试应贯穿在开发的各个阶段，测试过程基本与开发过程平行，测试目标来自于需求分析阶段。

【答案】A

5.7.2 传统软件的测试策略

1. 单元测试

单元测试也称为模块测试，在模块编写完成且无编译错误后就可以进行。如果选择机器测试，一般用白盒法，多个模块一起进行。单元测试主要检查模块的 5 个特征：模块接口、局部数据结构、重要的执行路径、出错处理、边界条件。提高模块的内聚度可以简化单元测试。单元测试的过程如图 5-9 所示。

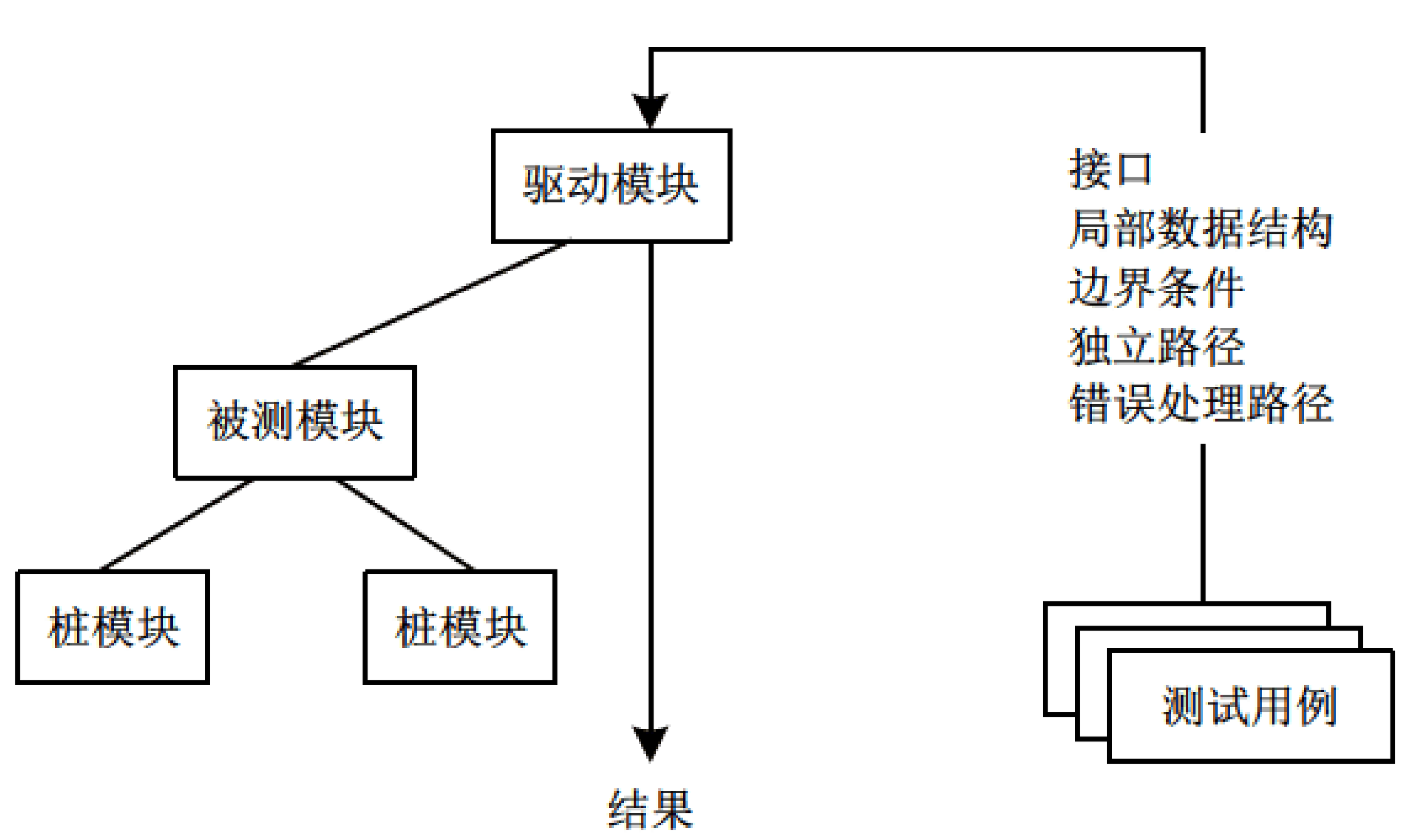


图 5-9　单元测试

2. 组装测试

组装测试也称为集成测试，就是把模块按系统设计说明书的要求组合起来进行测试。组装测试有两种方法：一种是分别测试各个模块，再把这些模块组合起来进行整体测试，即非增量式集成；另一种是把下一个要测试的模块组合到已测试好的模块中，测试完后再将下一个需要测试的模块组合起来，进行测试，逐步把所有模块组合在一起并完成测试，即增量式集成。

增量集成策略以下几种。

1)　自顶向下集成测试

模块的集成顺序为从主控模块开始，沿着控制层次逐步向下，以深度优先或广度优先的方式将从属于主控模块的模块集成到结构中。集成过程可以通过下列 5 个步骤完成。

(1)　主控模块用作测试驱动模块，用这些从属于主控模块的所有模块代替桩模块。

(2)　依靠所选择的集成方法(即深度优先或广度优先)，每次用实际模块替换一个从属桩模块。

(3)　在集成每个模块后都进行测试。

(4)　在完成每个测试集之后，用实际模块替换另一个桩模块。

(5)　可以执行回归测试，以确保没有引入新的错误。

2)　自底向上集成测试

从原子模块开始进行构造和测试。自底向上集成策略可以利用以下步骤来实现：

(1)　连接低层构件以构成完成特定子功能的簇。

(2)　编写驱动模块以协调测试用例的输入和输出。

(3)　测试簇。

(4)　去掉驱动程序，沿着程序结构向上逐步连接簇。

3)　回归测试

集成测试时，每加入一个新模块，软件就发生变更，这些变更可能使原来可以正常工作的功能产生问题。在集成测试策略的环境下，回归测试是重新执行已测试过的某些子集，以确保变更没有带来不期望的副作用。回归测试有助于保证变更不引入无意识行为或额外的错误。回归测试要执行的测试子集包含以下 3 种测试用例：

(1)　能够测试软件所有功能的具有代表性的测试样本。

(2)　额外测试，侧重于可能会受变更影响的软件功能。

(3)　侧重于已发生变更的软件构件测试。

4)　冒烟测试

本质上，冒烟测试方法包括下列活动：

(1)　将已经转换为代码的软件构件集成到构建中。(一个构建包括所有的数据文件库、可复用的模块，以及实现一个或多个产品功能所需的工程化构件)

(2)　设计一系列测试以暴露影响构建正确地完成其功能的错误，其目的是为了发现极有可能造成项目延迟的业务阻塞错误。

(3)　每天将该构建与其他构建及整个软件产品集成起来进行冒烟测试。

3. 确认测试

确认测试的任务是进一步检查软件的功能和性能是否与用户要求的一样。首先进行有效性测试以及软件配置审查，然后进行验收测试和安装测试，经过管理部门的认可和专家的鉴定后，软件即可交给用户使用。

若将软件开发为产品，由多个用户使用，让每个用户都进行正式的验收测试是不切实际的。多数软件开发者使用被称为 α 测试与 β 测试的过程，以期查找到似乎只有最终用户才能发现的错误。

4. 系统测试

系统测试是将已经确认的软件、计算机硬件、外设和网络等其他因素结合在一起，进行信息系统的各种组装测试和确认测试，其目的是通过与系统的需求相比较，发现所开发的系统与用户需求不符或矛盾的地方。常见的系统测试主要有恢复测试、安全性测试、压力测试、性能测试、部署测试。

真题链接

【例 5-11】单元测试中，检验模块接口时，不需要考虑__(36)__。(2013 年 5 月真题 36)

(36) A. 测试模块的输入参数和形式参数在个数、属性、单位上是否一致
B. 全局变量在各模块中的定义和用法是否一致
C. 输入是否改变了形式参数
D. 输入参数是否使用了尚未赋值或者尚未初始化的变量

【解析】在单元测试的任务中，检查模块接口时，需要考虑：①输入的实际参数与形式参数是否一致。②调用其他模块的实际参数与被调模块的形参是否一致。③全局变量的定义在各模块是否一致。④外部输入、输出等。因此 D 选项不在考虑范围之内。

【答案】D

5.7.3 测试方法

软件测试方法分为静态测试和动态测试。

1) 静态测试

静态测试是指被测试程序不在机器上运行，而是采用人工检测和计算机辅助静态分析的手段对程序进行检测。

2) 动态测试

动态测试是指通过运行程序发现错误。对软件产品进行动态测试时可以采用黑盒测试法和白盒测试法。

测试用例由测试输入数据和与之对应的预期输出结构组成。在设计测试用例时，应当包括合理的输入条件和不合理的输入条件。

1. 黑盒测试

黑盒测试也称为功能测试，在完全不考虑软件的内部结构和特性的情况下，测试软件的外部特性。

常用的黑盒测试技术有等价类划分、边界值分析、错误推测和因果图等。

① 等价类划分。等价类划分法将程序的输入域划分为若干等价类，然后从每个等价类中选取一个代表性数据作为测试用例。每一类的代表性数据在测试中的作用等价于这一类中的其他值。这样就可以用少量代表性的测试用例取得较好的测试效果。等价类划分有两种不同的情况：有效等价类和无效等价类。在设计测试用例时，要同时考虑这两种等价类。

② 边界值分析。输入的边界比中间更加容易发生错误，因此用边界值分析来补充等价类划分的测试用例设计技术。边界值分析选择等价类边界的测试用例，既注重于输入条件边界，又适用于输出域测试用例。

③ 错误推测。错误推测是基于经验和直觉推测程序中所有可能存在的错误，从而有针对性地设计测试用例的方法。其基本思想是列举出程序中所有可能有的错误和容易发生错误的特殊情况，根据它们选择测试用例。

④ 因果图。因果图法是从自然语言描述的程序规格说明中找出因(输入条件)和果(输出或程序状态的改变)，通过因果图转换为判定表。

2. 白盒测试

白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的路径和过程进行测试，检查是否满足设计的需要。

白盒测试常用的技术是逻辑覆盖、循环覆盖和基本路径测试。

① 逻辑覆盖。逻辑覆盖考查用测试数据运行被测程序时对程序逻辑的覆盖程度。主要的逻辑覆盖标准有语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖六种。

② 循环覆盖。执行足够的测试用例，使得循环中的每个条件都得到验证。

③ 基本路径测试。基本路径测试法是在程序控制流图的基础上，通过分析控制流图的环路复杂性，导出基本可执行路径集合，从而设计测试用例。

真题链接

【例 5-12】采用白盒测试方法对图 5-10 进行测试，设计了 4 个测试用例：①(x=0，y=3)，②(x=l, y=2)，③(x=−1，y=2)，④(x=3, y=l)。至少需要测试用例①②才能完成<u>(35)</u>覆盖，至少需要测试用例①②③或①②④才能完成<u>(36)</u>覆盖。(2014 年 5 月真题 35～36)

(35) A. 语句　　B. 条件　　C. 判定/条件　　D. 路径

(36) A. 语句　　B. 条件　　C. 判定/条件　　D. 路径

【解析】语句覆盖要求被测程序中的每一条语句至少执行一次，这种覆盖对程序执行逻辑的覆盖很低。

条件覆盖要求每一判定语句中每个逻辑条件的各种可能的值至少满足一次。判定/条件覆盖要求判定中每个条件的所有可能取值(真/假)至少出现一次，并使得每个判定本身的判定结果(真/假)也至少出现一次。路径覆盖则要求覆盖被测程序中所有可能的路径。

通过测试用例①(x=0，y=3)，能执行到语句 A，同时覆盖左侧路径；通过测试用例②(x=l, y=2)，能执

行到语句 B，同时覆盖右侧路径；通过测试用例③(x=−1，y=2)或④(x=3, y=l)，什么也不执行，覆盖中间路径。

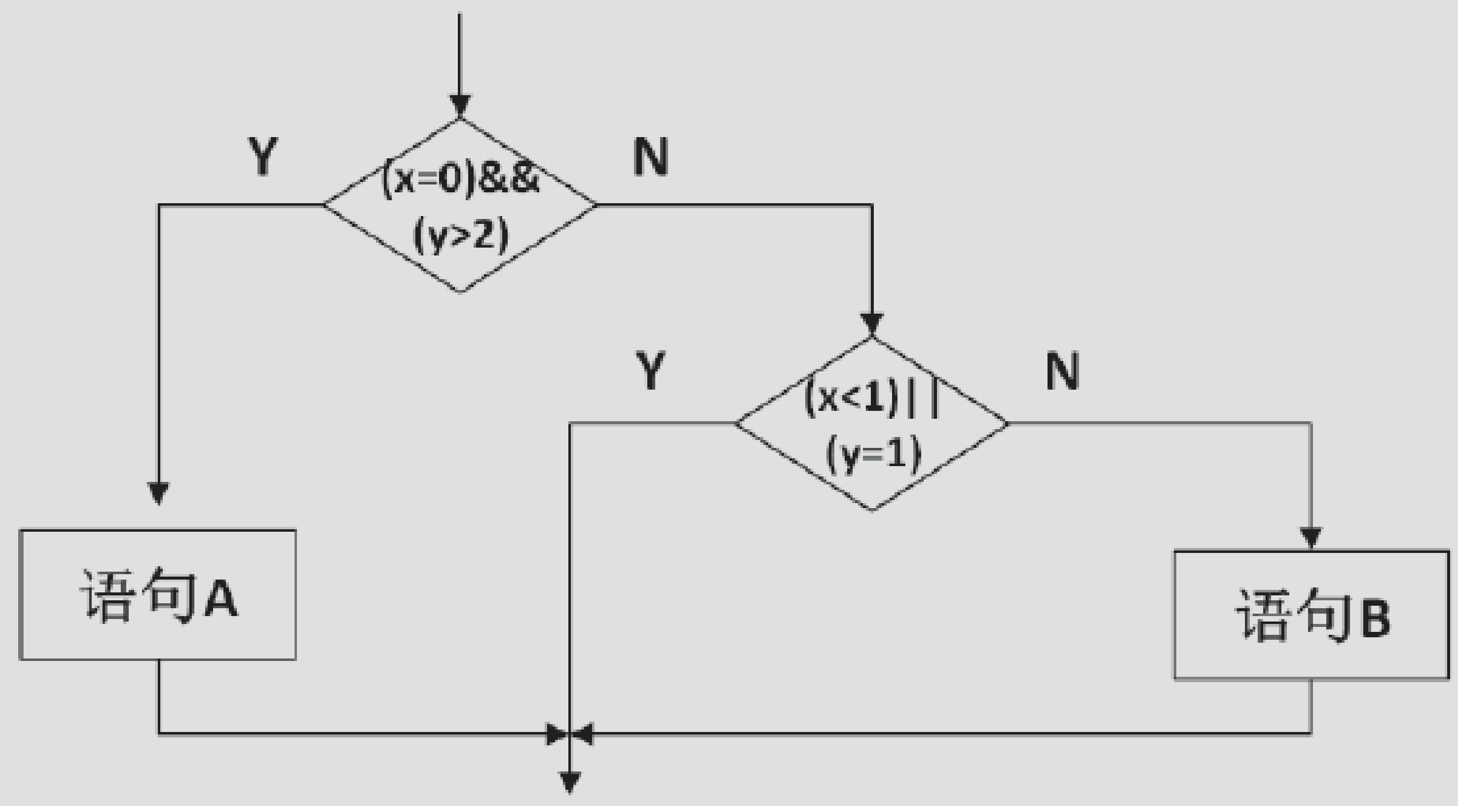


图 5-10　流程图

【答案】(35)A；(36)D

5.7.4　调试

调试的任务就是根据测试时所发现的错误，找出原因和具体的位置，进行改正。调试主要由程序开发人员来进行，谁开发的程序就由谁进行调试。常用的调试方法有试探法、回溯法、对分查找法、归纳法和演绎法。

5.7.5　测试 Web 应用系统

为了了解 Web 工程环境中的测试目标，必须考虑 WebApp 质量的多种维度。评估和测试都要检查下面质量维度中的一项或多项：内容、功能、结构、可用性、导航性、性能、兼容性、安全性。

WebApp 测试策略采用所有软件测试使用的基本原理，并建议使用面向对象系统使用的策略和战术。

5.8　系统运行和维护知识

考核说明：本节主要介绍系统转换、系统可维护性的概念、系统维护的内容和类型、系统评价等内容，在考试中经常会考到软件维护的内容，要重点掌握。

5.8.1　系统转换

新系统运行成功之后，就可以在新系统和老系统之间互相转换。新老系统之间的转换方式有直接转换、并行转换和分段转换。

(1)　直接转换：在确定新系统运行无误后，立刻启用新系统，终止老系统运行。

(2)　并行转换：新老系统并行一段时间，经过一段时间的考验以后，新系统正式代替

老系统。

(3) 分段转换：又称逐步转换、向导转换、试点过渡法等。这种方法实际上是以上两种转换方式的结合。

5.8.2　系统维护概述

1. 系统可维护性的概念

系统可维护性定义为：维护人员理解、改正、改动和改进这个软件的难易程度。提高可维护性是开发管理信息系统所有步骤的关键目的，系统是否能够被很好地维护，可以用系统的可维护性这一指标来衡量。

(1) 系统的可维护性指标：可理解性、可测试性和可修改性。

(2) 软件文档：文档是软件可维护性的决定因素。软件系统的文档可以分为用户文档和系统文档两类。

(3) 软件文档的修改：每当改动数据、软件结构、模块过程或任何其他有关的软件特点时，必须立即修改相应的技术文档。

2. 系统维护的内容及类型

1) 硬件维护

硬件维护应由专职的硬件维护人员来负责，主要有两种类型的维护活动，一种是定期的设备保养性维护，另一种是突发性的故障维护。

2) 软件维护

软件维护主要是根据需求变化或硬件环境的变化对应用程序进行部分或全部的修改。软件维护的内容包括正确性维护、适应性维护、完善性维护和预防性维护等。

(1) 正确性维护。在软件交付使用后，必然会有一部分隐藏的错误被带入运行阶段，这些隐藏的错误在某些特定的使用环境下就会暴露出来。为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应当进行的诊断和改正错误的过程，就叫作正确性维护。

(2) 适应性维护。随着计算机的飞速发展，外部环境(新的硬、软件配置)或数据环境(数据库、数据格式、数据输入/输出方式、数据存储介质)可能发生变化，为了使软件适应这种变化，而去修改软件的过程就叫作适应性维护。

(3) 完善性维护。在软件的使用过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动叫作完善性维护。

(4) 预防性维护。为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的变化，应主动增加预防性的新功能，以使应用系统适应各类变化而不被淘汰。例如将专用报表功能改成通用报表生成功能，以适应将来报表格式的变化。这方面的维护工作量占整个维护工作量的 4%左右。

3) 数据维护

数据维护主要是由数据库管理员来负责，主要负责数据库的安全性和完整性以及进行并发性控制。

3. 系统维护的管理和步骤

系统维护的管理和步骤如下。

(1) 提出维护或修改要求。

(2) 领导审查并做出答复，如同意修改则列入维护计划。

(3) 领导分配任务，维护人员修改。

(4) 验收维护成果并登记修改信息。

真题链接

【例 5-13】以下关于软件维护和可维护性的叙述中，不正确的是 (36) 。(2014 年 11 月真题 36)

(36) A. 软件维护要解决软件产品交付用户之后运行中发生的各种问题

B. 软件的维护期通常比开发期长得多，其投入也大得多

C. 进行质量保证审查可以提高软件产品的可维护性

D. 提高可维护性是在软件维护阶段考虑的问题

【解析】在软件设计阶段就应该考虑软件的可维护性。

【答案】D

【例 5-14】某搜索引擎在使用过程中，若要增加接受语音输入的功能，使得用户可以通过语音输入来进行搜索，此时应对系统进行 (34) 维护。 (2014 年 5 月真题 34)

(34) A. 正确性　　B. 适应性　　C. 完善性　　D. 预防性

【解析】在软件设计阶段就应该考虑软件的可维护性。

【答案】C

5.8.3 系统评价

1. 系统评价概述

信息系统的评价分为广义和狭义两种。广义的信息系统评价是指从系统开发开始到结束的每一阶段都需要进行评价。狭义的信息系统评价则是指在系统建成并投入运行之后所进行的全面、综合的评价。

按评价的时间与信息系统所处的阶段的关系，又可从总体上把广义的信息系统评价分成立项评价、中期评价和结项评价。

2. 系统评价的指标

从以下几方面综合考虑，建立起一套指标体系理论框架。

(1) 从信息系统的组成部分出发，信息系统是一个由人机共同组成的系统，所以可以按照运行效果和用户需求(人)、系统质量和技术条件(机)这两条线索构造指标。

(2) 从信息系统的评价对象出发，对于开发方来说，他们所关心的是系统质量和技术

水平；对于用户方而言，关心的是用户需求和运行质量；系统外部环境则主要通过社会效益指标来反映。

(3) 从经济学角度出发，分别按系统成本、系统效益和财务指标三条线索建立指标。

5.9　应试加油站

5.9.1　考频统计

通过对近几年考试真题的分析，“系统开发和运行”模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 5-1 所示。

表 5-1　历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：15～16、32、36	结构化开发方法、模块、软件维护、测试用例	4
	下午：试题一	数据流图(DFD)的应用	15
2014 年 5 月	上午：15～16、34～36	结构化开发方法、模块功能内聚、系统测试、软件维护	5
	下午：试题一	数据流图(DFD)的应用	15
2013 年 11 月	上午：15～16、34～36	数据流图、模块化、软件维护、软件测试	5
	下午：试题一	数据流图(DFD)的应用	15
2013 年 5 月	上午：19、29、33、33、36	数据流图、回归测试、软件维护、测试用例、单元测试	5
	下午：试题一	数据流图(DFD)的应用	15
2012 年 11 月	上午：15、34～36	数据流图、软件维护、测试用例	4
	下午：试题一	数据流图(DFD)的应用	15
2012 年 5 月	上午：15～16、18、31、34～35	数据流图、模块内聚、人员组织、软件维护、白盒测试	6
	下午：试题一	数据流图(DFD)的应用	15

纵观历年试卷，本章知识点是以选择题和综合分析题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量大约为 5 道选择题，所占分值为 5 分(约占试卷总分值 75 分中的 7%)；在下午试卷中，所考查的题量为 1 道综合分析题，所占分值为 15 分(约占试卷总分值 75 分中的 20%)。本章试题理论与实践应用并重，难度为中等偏难。模块、系统测试、系统维护是上午考试的重点，数据流图的应用是下午必考的内容，要重点掌握。

5.9.2 解题技巧

【典型题 5-1】以下关于结构化开发方法的叙述中，不正确的是<u>(15)</u>。(2014 年 5 月真题 15)

(15) A. 将数据流映射为软件系统的模块结构

B. 一般情况下，数据流类型包括变换流型和事务流型

C. 不同类型的数据流有不同的映射方法

D. 一个软件系统只有一种数据流类型

【解析】结构化分析与设计方法是一种面向数据流的开发方法，它以数据流为中心构建软件的分析模型和设计模型。结构化设计是将结构化分析的结构(数据流图)映射成软件的体系结构(结构图)。根据信息流的特点，可将数据流图分为变换型数据流图和事务型数据流图，其对应的映射分别称为变换分析和事务分析。

在规模较大的实际系统中，其数据流图往往是变换型和事务型的混合结构，此时可以把变换分析和事务分析应用在同一数据流图的不同部分。

【答案】D

【典型题 5-2】以下关于文档的叙述中，不正确的是<u>(33)</u>。(2012 年 11 月真题 33)

(33) A. 文档仅仅描述和规定了软件的使用范围及相关的操作命令

B. 文档是软件产品的一部分，没有文档的软件不能称为完整的软件

C. 软件文档的编制在软件开发工作中占有突出的地位和相当大的工作量

D. 高质量文档对于发挥软件产品的效益有着重要的意义

【解析】信息系统的文档，是系统建设过程的“痕迹”，是系统维护人员的指南，是开发人员与用户交流的工具。信息系统的文档，不但包括应用软件开发过程中产生的文档，还包括硬件采购和网络设计中形成的文档。软件文档的编制贯穿于软件产品开发的各个阶段，是提高软件产品开发效率、规范软件产品开发过程、保证软件产品质量的关键。

【答案】A

【典型题 5-3】在如图 5-11 所示的数据流图中，共存在<u>(29)</u>个错误。(2013 年 5 月真题 29)

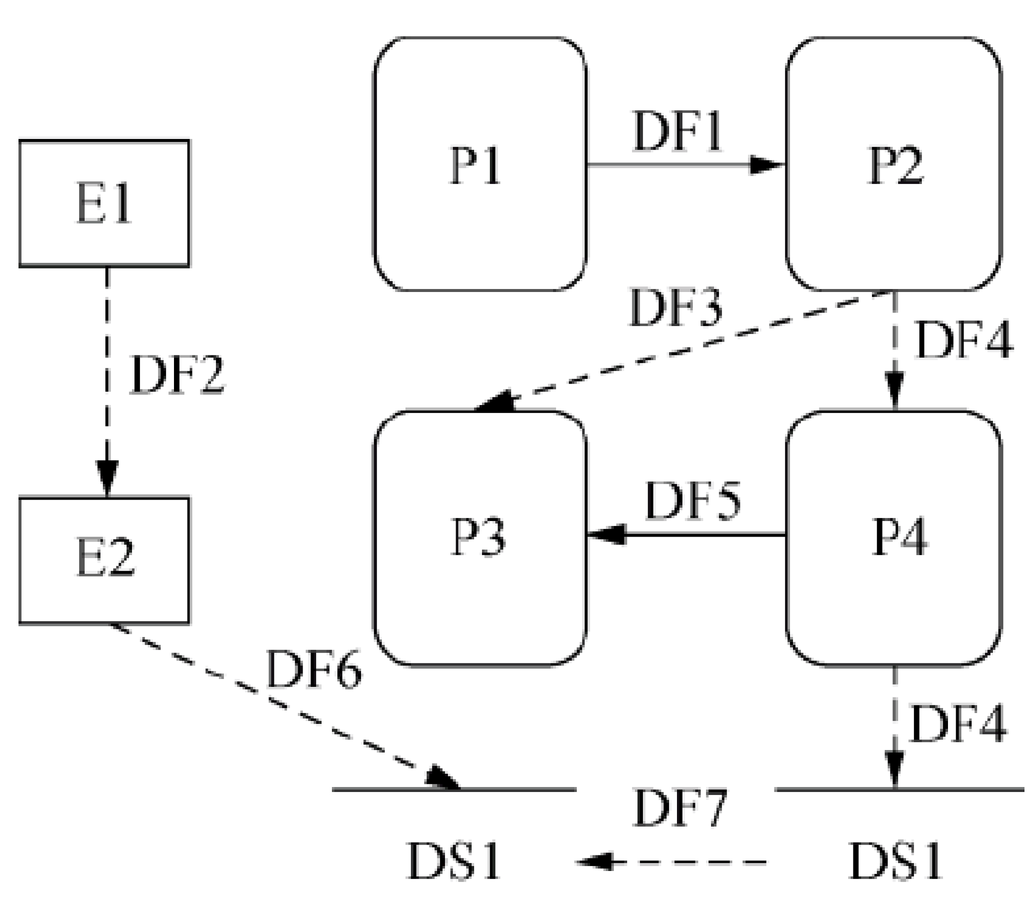


图 5-11 数据流图

(29) A. 4　　B. 6　　C. 8　　D. 9

【解析】数据流图有如下几个设计原则。

① 数据守恒原则：对任何一个加工来说，其所有输出数据流中的数据必须能从该加工的输入数据流中直接获得，或者说是通过该加工能产生的数据。

② 守恒加工原则： 对同一个加工来说，输入与输出的名字必须不相同，即使它们的组成成分相同。

③ 对于每个加工，必须既有输入数据流，又有输出数据流。

④ 外部实体与外部实体之间不存在数据流。

⑤ 外部实体与数据存储之间不存在数据流。

⑥ 数据存储与数据存储之间不存在数据流。

⑦ 父图与子图的平衡原则：子图的输入输出数据流同父图相应加工的输入输出数据流必须一致，此即父图与子图的平衡。父图与子图之间的平衡原则不存在于单张图。

⑧ 数据流与加工有关，且必须经过加工。

根据以上这些原则，DF2 违背了原则 4，DF6 违背了原则 5，DF7 违背了原则 6，P1 和 P3 缺少数据流，违背了原则 3，P4 的输入输出数据流相同，违背了原则 2。因此共有 6 个错误。

【答案】 B

【典型题 5-4】 以下关于数据流图的叙述中，不正确的是 (15) 。(2012 年 11 月真题 15)

(15) A. 每条数据流的起点和终点必须是加工

B. 必须保持父图与子图平衡

C. 每个加工必须有输入数据流，但可以没有输出数据流

D. 应保持数据守恒

【解析】 加工描述的是输入数据流到输出数据流之间的变换，也就是输入数据流经过什么样处理后变成了输出数据流。因此，每个加工必须既要有输入数据流，又要有输出数据流。

【答案】 C

【典型题 5-5】 数据流图(DFD)对系统的功能和功能之间的数据流进行建模，其中顶层数据流图描述了系统的 (15) 。(2012 年 5 月真题 15)

(15) A. 处理过程　　B. 输入和输出　C. 数据存储　　D. 数据实体

【解析】 顶层数据流图只含有一个加工表示整个系统，输出数据流和输入数据流为系统的输入数据和输出数据，表明系统的范围，以及与外部环境的数据交换关系。主要明确系统从哪些外部实体接收数据流，以及系统发送数据流到哪些外部实体。

【答案】 B

【典型题 5-6】 在划分模块时，一个模块的作用范围应该在其控制范围之内。若发现其作用范围不在其控制范围内，则 (16) 不是适当的处理方法。(2013 年 11 月真题 16)

(16) A. 将判定所在模块合并到父模块中，使判定处于较高层次

B. 将受判定影响的模块下移到控制范围内

C. 将判定上移到层次较高的位置

D. 将父模块下移，使该判定处于较高层次

【解析】 模块的控制范围包括该模块本身及所有的下属模块的集合。控制范围完全取决于系统的结构，当影响范围超出控制范围时，可将判定所在模块合并到父模块中，使判定处于较高层次。所以本题答案选 D。

【答案】 D

【典型题 5-7】 模块 A 执行几个逻辑上相似的功能，通过参数确定该模块完成哪一个功能，则该模块具有 (16) 内聚。(2012 年 5 月真题 16)

(16) A. 顺序　　B. 过程　　C. 逻辑　　D. 功能

【解析】内聚有偶然内聚、逻辑内聚、时间内聚、通信内聚、顺序内聚和功能内聚。逻辑内聚指模块内执行几个逻辑上相似的功能，通过参数确定该模块完成哪一个功能。顺序内聚指一个模块中各个处理元素都密切相关于同一功能且必须顺序执行，前一功能元素的输出就是下一功能元素的输入。功能内聚是指模块内所有元素共同完成一个功能，缺一不可，是最强的内聚。

【答案】C

【典型题 5-8】 __(18)__ 最不适宜采用无主程序员组的开发人员组织形式。(2012 年 5 月真题 18)

(18) A. 项目开发人数少(如 3～4 人)的项目　　B. 采用新技术的项目

C. 大规模项目　　D. 确定性较小的项目

【解析】无主程序员组中的成员相互平等，工作目标和决策都由全体成员民主讨论。这种组有利于发挥每个人的积极性，但往往职责不明，出了问题谁也不负责，而且不利于与外界的联系。很显然，大规模的项目不适合采用这种人员组织形式。

【答案】C

【典型题 5-9】某项目为了修正一个错误而进行了修改。错误修改后，还需要进行 __(19)__ 以发现这一修改是否引起原本正确运行的代码出错。(2013 年 5 月真题 19)

(19) A.单眼测试　　B. 接收测试　　C.安装测试　　D. 回归测试

【解析】回归测试是指修改了旧代码后，重新进行测试以确认修改没有引入新的错误或导致其他代码产生错误。

【答案】D

【典型题 5-10】在软件维护中，由于企业的外部市场环境和管理需求的变化而导致的维护工作，属于 __(33)__ 维护。(2013 年 5 月真题 33)

(33) A. 正确性　　B. 适应性　　C. 完善性　　D. 预防性

【解析】软件维护包括：①改正性维护，是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。②适应性维护，是指使用软件适应信息技术变化和管理需求变化而进行的修改。③完善性维护，这是为扩充功能和改善性能而进行的修改，主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。④预防性维护，为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的变化，应主动增加预防性的新的功能，以使应用系统适应各类变化而不被淘汰。显然该题中的情况属于适应性维护。

【答案】B

【典型题 5-11】在设计测试用例时，应遵循 __(35)__ 原则。(2013 年 5 月真题 35)

(35) A. 仅确定测试用例的输入数据，无须考虑输出结果

B. 只需检验程序是否执行应有的功能，不需要考虑程序是否做了多余的功能

C. 不仅要设计有效合理的输入，也要包含不合理、失效的输入

D. 测试用例应设计得尽可能复杂

【解析】测试用例要包括欲测试的功能、应输入的数据和预期的输出结果。测试数据应该选用少量、高效的测试数据进行尽可能完备的测试；基本目标是：设计一组发现某个错误或某类错误的测试数据。测试用例应覆盖有效合理的输入，也要包括不合理、失效的输入。

【答案】C

【典型题 5-12】逆向工程从源代码或目标代码中提取设计信息，通常在原软件生命周期的 __(34)__ 阶

段进行。(2013 年 11 月真题 34)

(34) A. 需求分析　　B. 软件设计　　C. 软件实现　　D. 软件维护

【解析】逆向工程是根据已有的东西和结果，通过分析来推导出具体的实现方法。逆向软件工程的目的是使软件得以维护。

【答案】D

【典型题 5-13】一个程序根据输入的年份和月份计算该年中该月的天数，输入参数包括年份(正整数)、月份(用 1～12 表示)。若用等价类划分测试方法进行测试，则 (35) 不是一个合适的测试用例(分号后表示测试的输出)。(2013 年 11 月真题 35)

(35) A. (2013,1;31)　　B. (0,1;'错误')　　C. (0,13;'错误')　　D. (2000,−1; '错误')

【解析】等价类划分测试方法是把所有可能输入的数据，即程序的输入域划分成若干部分(子集)，然后从每一个子集中选取少数具有代表性的数据作为测试用例。该方法是一种重要的、常用的黑盒测试用例设计方法。答案 C 不是一个合格的测试用例。

【答案】C

【典型题 5-14】 (36) 不是单元测试主要检查的内容。(2013 年 11 月真题 36)

(36) A. 模块接口　　B. 局部数据结构

C. 全局数据结构　　D. 重要的执行路径

【解析】单元测试主要检查的内容包括：边界测试、错误处理测试、路径测试、局部数据结构测试、模块接口测试。所以答案选 C。

【答案】C

【典型题 5-15】由于信用卡公司升级了其信用卡支付系统，导致超市的原有信息系统也需要做相应的修改工作，该类维护属于 (34) 。(2012 年 11 月真题 34)

(34) A. 正确性维护　　B. 适应性维护　　C. 完善性维护　　D. 预防性维护

【解析】正确性维护是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误；适应性维护是指应用软件适应信息技术变化和管理需求变化而进行的修改；完善性维护是为扩充功能和改善性能而进行的修改，主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能和性能特征。预防性维护是指为了改进应用软件的可靠性和可维护性，为了适应未来的软硬件环境的变化，应主动增加预防性的新功能，以使应用系统适应各种变化而不被淘汰。本例的系统是适应信用卡支付系统的升级而做出的修改，属于适应性维护。

【答案】B

【典型题 5-16】用白盒测试方法对图 5-12 所示的程序进行测试，设计了 4 个测试用例：①(x=0,y=3)，②(x=1,y=2)，③(x=−1,y=2)，④(x=3,y=1)。测试用例①②实现了 (35) 覆盖；若要完成路径覆盖，则可用测试用例 (36) 。(2012 年 11 月真题 35～36)

(35) A. 语句　　B. 条件　　C. 判定　　D. 路径

(36) A. ①②　　B. ②③　　C. ①②③　　D. ①③④

【解析】白盒测试也称为结构测试，根据程序的内部结构和逻辑来设计测试用例，对程序的路径和过程进行测试，检查是否满足设计的需要。语句覆盖是指选择足够的测试数据，使被测试程序中每条语句至少执行一次。(x=0,y=3)的测试用例使得语句 A 被执行，(x=1,y=2)的测试用例使得语句 B 被执行。因此，测试用例①②实现了语句覆盖。

路径覆盖是指覆盖被测试程序中所有可能的路径。图 5-12 中包含 4 条路径，①覆盖了最左边的路径，

②覆盖了第二和最右边的两条路径，③覆盖了使(x<1)||(y=1)成立的路径。因此使用①②③测试用例，可完成路径覆盖。

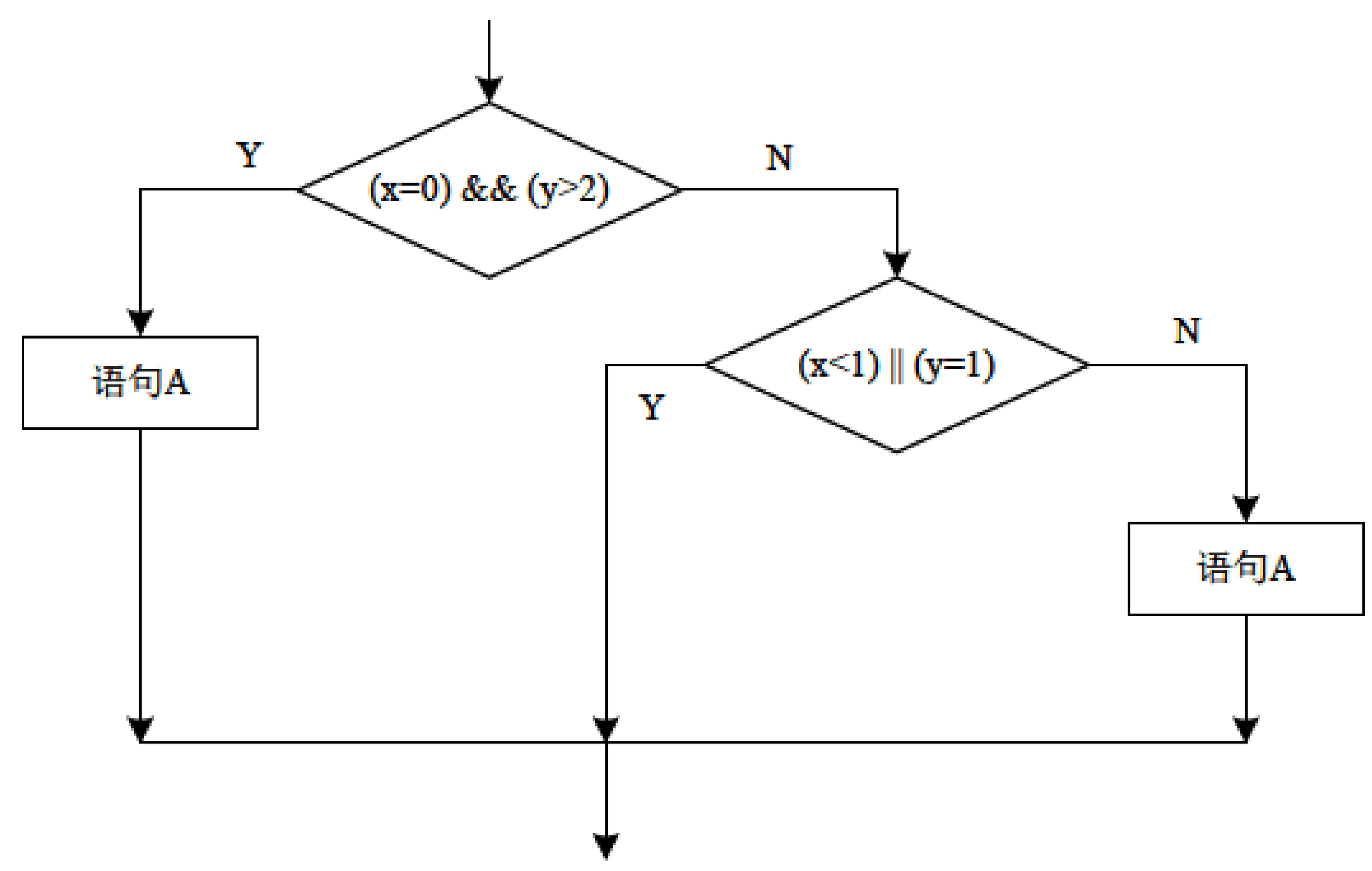


图 5-12　流程图

【答案】(35)A；(36)C

【典型题 5-17】某企业由于外部市场环境和管理需求的变化对现有软件系统提出新的需求，则对该软件进行的维护属于__(31)__维护。(2012 年 5 月真题 31)

(31) A. 正确性　　B. 完善性　　C. 适应性　　D. 预防性

【解析】适应性维护是指应用软件适应信息技术变化和管理需求变化而进行的修改。如改善系统硬件环境而产生系统更新换代的需求，企业由于外部市场环境和管理需求的变化对现有软件系统提出新的需求等。

正确性维护是指改正系统开发阶段已发生而系统测试阶段尚未发现的错误。

完善性维护主要是指对已有的软件系统增加一些在系统分析和设计阶段中都没有规定的功能和性能特征。

预防性维护则是为了适应未来的软硬件环境的变化，主动增加预防性的新的功能，以使应用系统适应各类变化而不被淘汰。

【答案】C

【典型题 5-18】在白盒测试法中，__(34)__是最弱的覆盖准则。图 5-13 至少需要__(35)__个测试用例才可以完成路径覆盖，语句组 2 不对变量 i 进行操作。(2012 年 5 月真题 34、35)

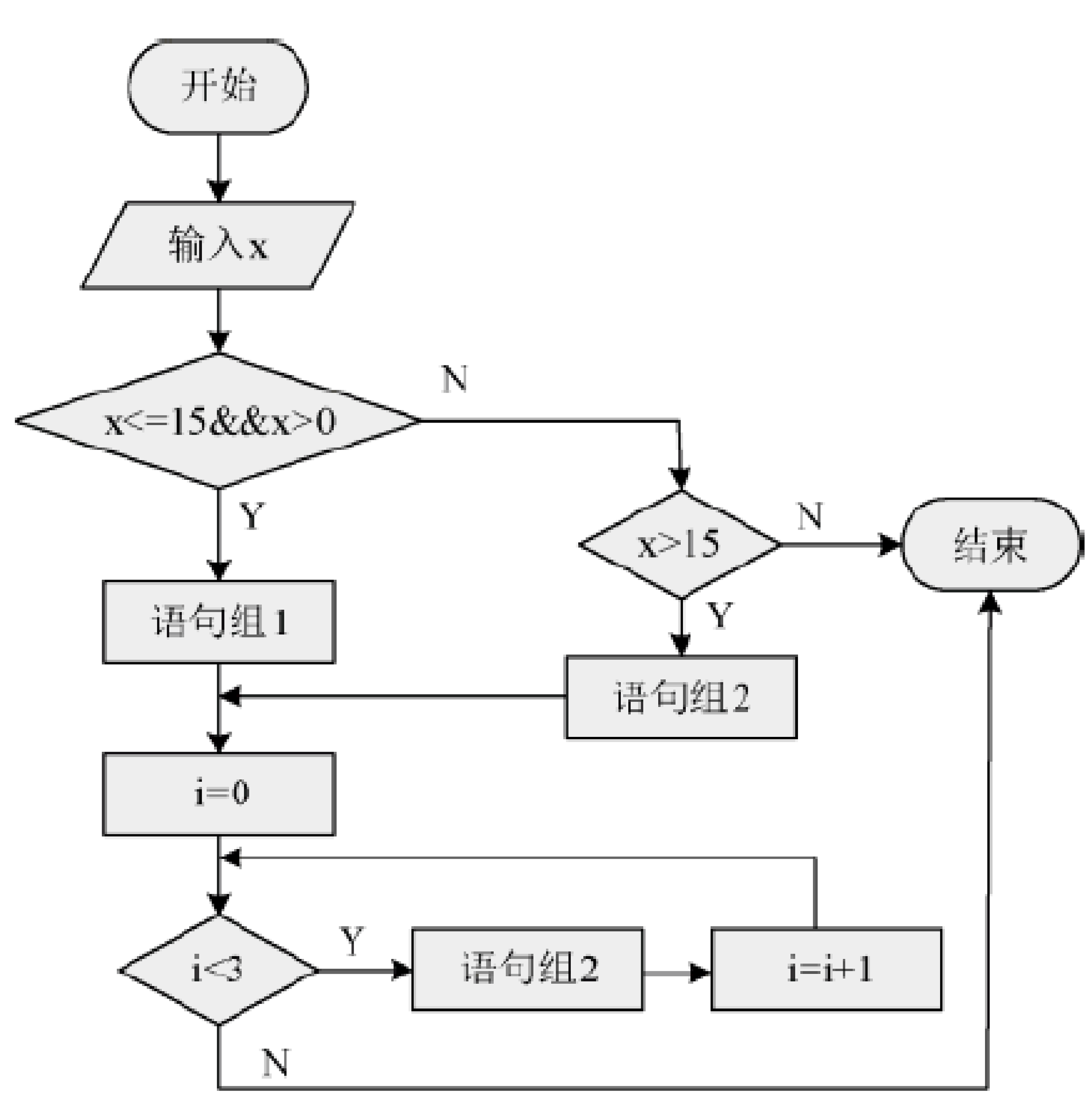


图 5-13　流程图

(34) A. 语句　　B. 条件　　C. 判定　　D. 路径

(35) A. 1　　B. 2　　C. 3　　D. 4

【解析】从覆盖源程序语句的详尽程度分析，逻辑覆盖标准包括以下不同的覆盖标准：语句覆盖、判定覆盖、条件覆盖、判定/条件组合覆盖、条件组合覆盖和路径覆盖。语句覆盖的含义是：选择足够多的测试数据，使被测程序中每条语句至少执行一次。语句覆盖是很弱的逻辑覆盖。

路径覆盖要求设计足够的测试用例，覆盖程序中所有可能的路径。路径覆盖是最强的逻辑覆盖。对于比较简单的小程序来说，实现路径覆盖是可能的，但是如果程序中出现了多个判断和多个循环，可能的路径数目将会急剧增长，以致实现路径覆盖是几乎不可能的。所以我们需要路径分析，计算程序中的路径数(复杂度)。公式如下：

$$V(G)=e-n+2$$

其中，e为边数，n为节点数。

本题，e=13，n=11，则路径数为4。而题目，语句组2不对变量i进行操作，因此边数e-1，路径数为3。

【答案】(34)A；(35)C

【典型题 5-19】阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。(2013 年 5 月试题一)

【说明】

某慈善机构欲开发一个募捐系统，已跟踪记录为事业或项目向目标群体进行募捐而组织的集体性活动。该系统的主要功能如下所述。

(1) 管理志愿者。根据募捐任务给志愿者发送加入邀请、邀请跟进、工作任务；管理志愿者提供邀请响应、志愿者信息、工作时长、工作结果等。

(2) 确定募捐需求和收集所募捐赠(资金及物品)。根据需求提出募捐任务、活动请求和捐赠请求，获取所募集的资金和物品。

(3) 组织募捐活动。根据活动请求，确定活动时间范围。根据活动时间，搜索场馆，即：向场馆发送场馆可用性请求，获得场馆可用性。然后根据活动时间和地点推广募捐活动，根据相应的活动信息举办活动，从募捐机构获取资金并向其发放赠品。获取和处理捐赠，根据捐赠请求，提供所募集的捐赠；处理与捐赠人之间的交互，即：录入捐赠人信息，处理后存入捐赠人信息表；从捐赠人信息表中查询捐赠人信息，向捐赠人发送募捐请求，并将已联系的捐赠人存入已联系的捐赠人表。根据捐赠请求进行募集，募得捐赠后，将捐赠记录存入捐赠表；对捐赠记录进行处理后，存入已处理捐赠表，向捐赠人发送致谢函，根据已联系的捐赠人和捐赠记录进行跟踪，将捐赠跟进情况发送给捐赠人。

先采用结构化方法对募捐系统进行分析与设计，获得如图 5-14 至图 5-16 所示分层数据流图。

【问题 1】使用说明中的词语，给出图 5-14 中的实体 E1～E4 的名称。

【问题 2】在建模 DFD 时，需要对有些复杂加工(处理)进行进一步精化，图 5-15 为图 5-14 中处理 3 的进一步细化的 1 层数据流图，图 5-16 为图 5-15 中 3.1 进一步细化的 2 层数据流图。补全图 5-15 中加工 P1、P2 和 P3 的名称和图 5-15 与图 5-16 中缺少的数据流。

【问题 3】使用说明中的词语，给出图 5-16 中的数据存储 D1～D4 的名称。

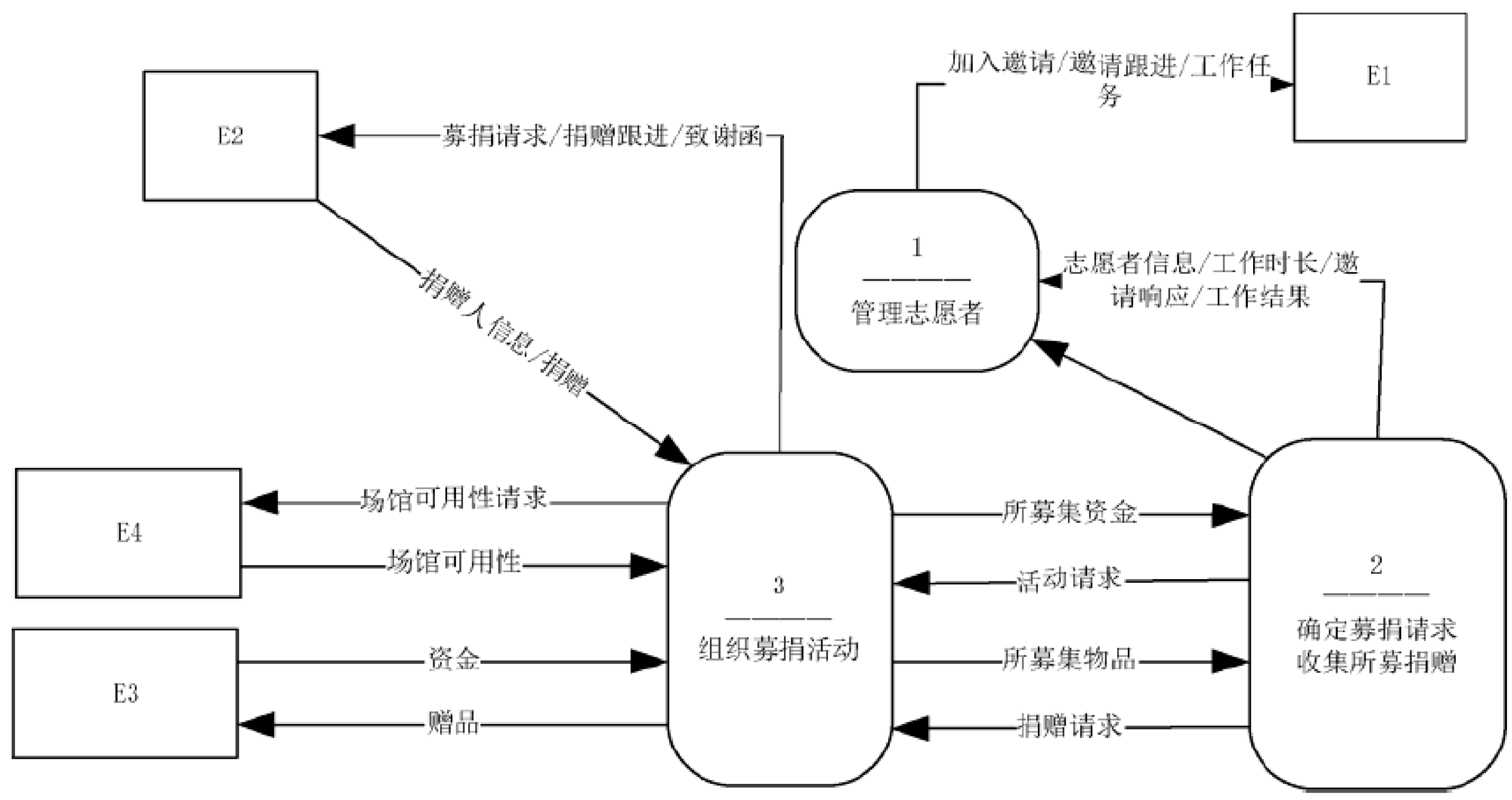


图 5-14 0 层数据流图

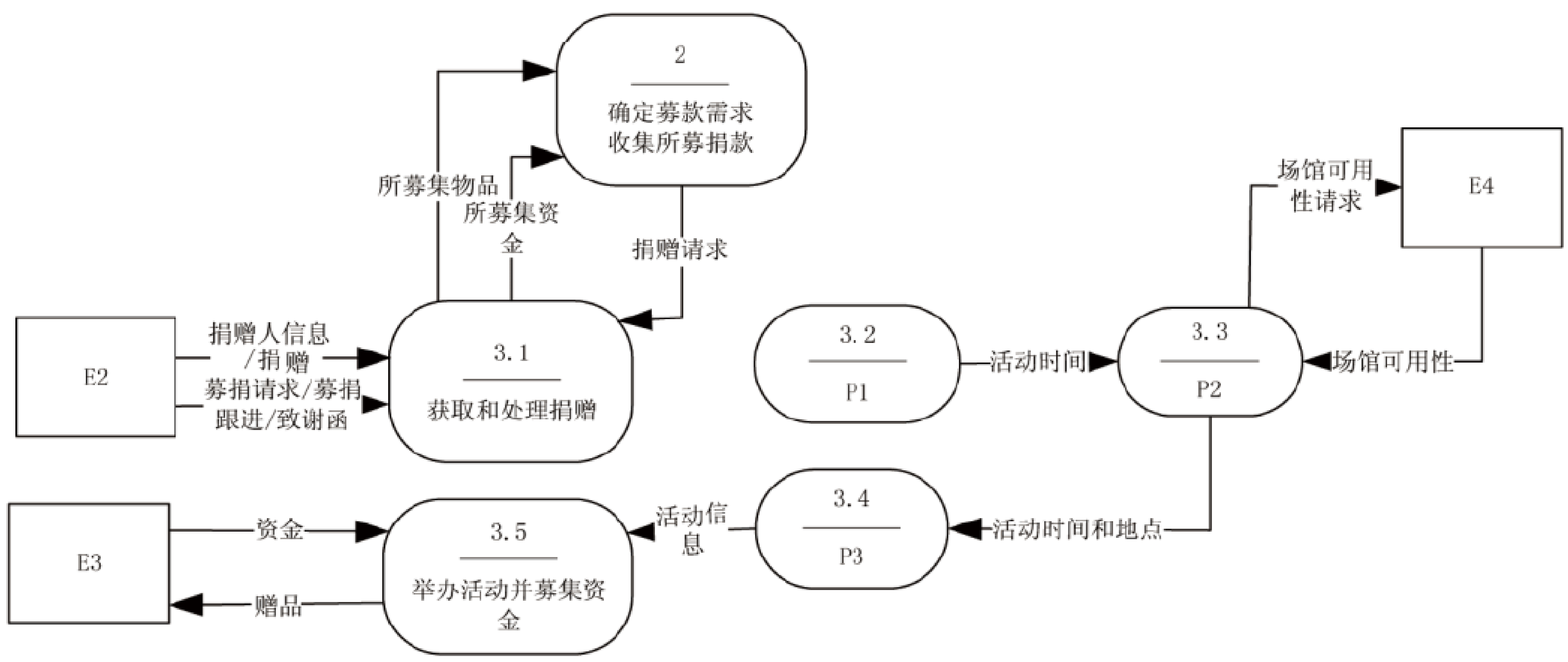


图 5-15 1 层数据流图

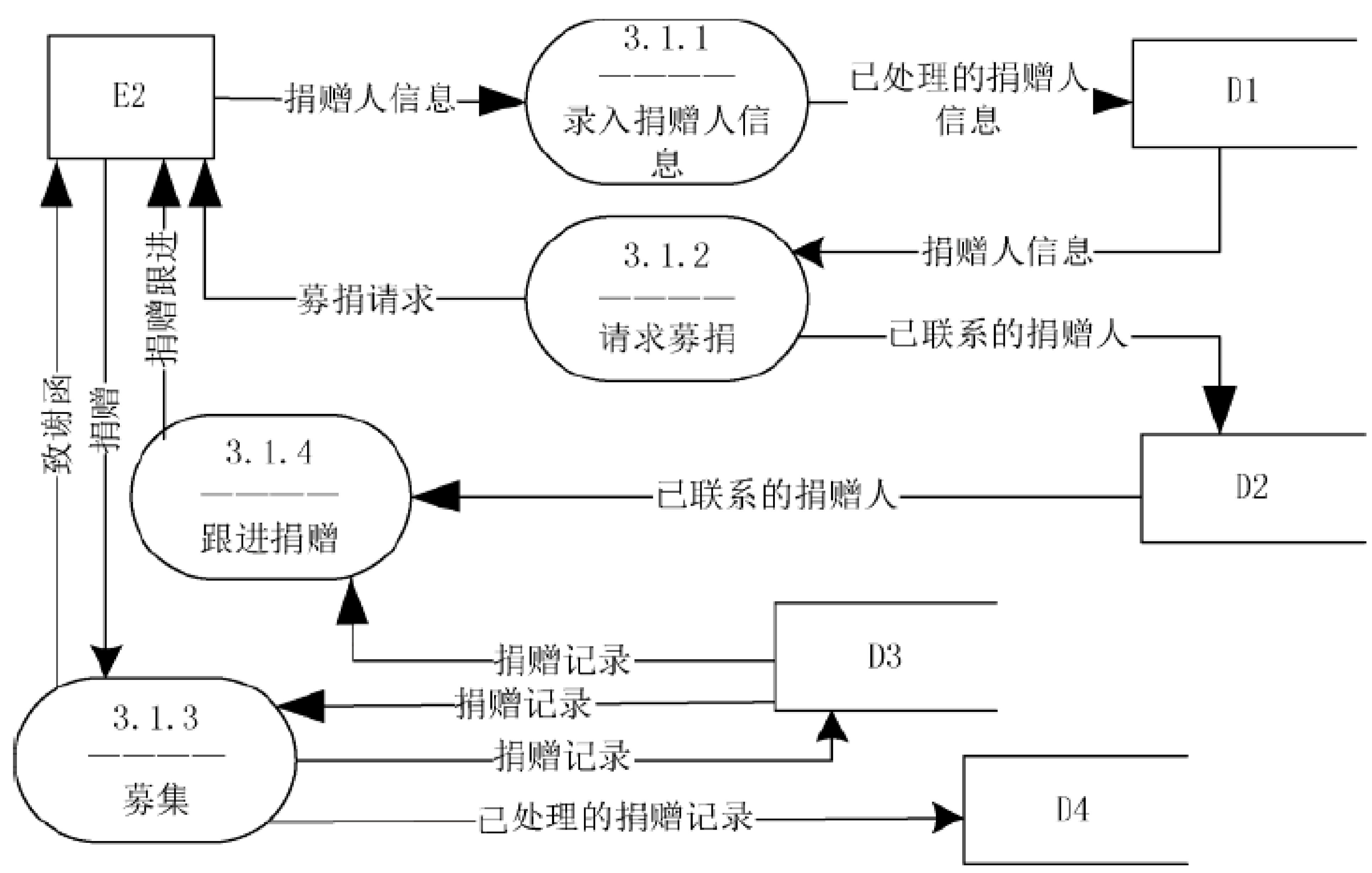


图 5-16 2 层数据流图

【解析】

该题以募捐系统为载体来考核考生对数据流图知识点的掌握程度。从题目的问答形式上来看，和往年相似，要求补充外部实体、补充缺失数据流、找出外部存储。解答这类问题，有两个原则：

①第一个原则是紧扣试题系统说明部分，数据流图与系统说明有着严格的对应关系，系统说明部分的每一句话都能对应到图中来，解题时可以一句一句地对照图来分析。

②第二个原则即数据的平衡原则，这一点在解题过程中也是至关重要的。数据平衡原则有两方面的含义，一方面是分层数据流图父子图之间的数据流平衡原则，另一方面是每张数据流图中输入与输出数据流的平衡原则。

【问题 1】根据 0 层数据流管理志愿者中的募捐任务给志愿者发送加入邀请，邀请跟进，工作任务和管理志愿者提供的邀请响应等可知 E1 为志愿者；从录入捐赠人信息，向捐赠人发送募捐请求，向捐赠人发送致谢函等可知 E2 为捐赠人；根据说明中从募捐机构获取资金并向其发放赠品可知 E3 为募捐机构；根据向场馆发送可用性请求和获得场馆可用性可知 E4 为场馆。

【问题 2】根据 1 层数据流图中 P1 的输出流活动时间再结合说明可知 P1 为确定活动时间范围；从加工 P2 的输入流活动时间和输出流场馆可用性请求和活动时间和地点可知 P2 为搜索场馆；题目的说明中指出，根据活动时间和地点推广募捐活动，根据相应的活动信息举办活动，再结合 P3 的输入输出流可知 P3 为推广募捐活动。比较 0 层和 1 层中的数据流可知，P1 加工只有输出流，故缺少输入流，根据说明可知需要根据活动请求才能确定 P1，故该数据流为活动请求，在 0 层数据流中活动请求的起始加工为确定募款需求收集所募捐赠，故可知答案。

【问题 3】根据最后的说明和 2 层数据流可知 D1 为捐赠人信息表，D2 为已联系的捐赠人信息表，D3 为捐赠表，D4 为已经处理的捐赠表。

【答案】

【问题 1】E1：志愿者；E2：捐赠人；E3：募捐机构；E4：场馆

【问题 2】P1：确定活动时间范围；P2：搜索场馆；P3：推广募捐活动

缺少的数据流：

名称：活动请求；起点：确定募款需求收集所募捐赠；终点：P1

【问题 3】D1：捐赠人信息表；D2：已联系的捐赠人信息表；D3：捐赠表；D4：已经处理的捐赠表

【典型题 5-20】阅读以下说明和图，根据要求回答问题 1～问题 4。(2013 年 11 月试题一)

【说明】

某大学欲开发一个基于 Web 的课程注册系统，该系统的主要功能如下。

1. 验证输入信息

(1) 检查学生信息：检查学生输入的所有注册所需信息。如果信息不合法，返回学生信息不合法提示；如果合法，输出合法学生信息。

(2) 检查学位考试结果：检查学生提供的学位考试结果。如果不合法，返回学位考试结果不合法提示；如果合法，检查该学生注册资格。

(3) 检查学生注册资格：根据合法学生信息和合法学位考试结果，检查该学生对欲选课程的注册资格。如果无资格，返回无注册资格提示；如果有注册资格，则输出注册学生信息(包含选课学生标识)和欲注册课程信息。

2. 处理注册申请

(1) 存储注册信息：将注册学生信息记录在学生库。

(2) 存储所注册课程：将选课学生标识与欲注册课程进行关联，然后存入课程库。

(3) 发送注册通知：从学生库中读取注册学生信息，从课程库中读取所注册课程信息，给学生发送接收提示；给教务人员发送所注册课程信息和已注册学生信息。

现采用结构化方法对课程注册系统进行分析与设计，获得如图 5-17 所示的 0 层数据流图和图 5-18 所示的 1 层数据流图。

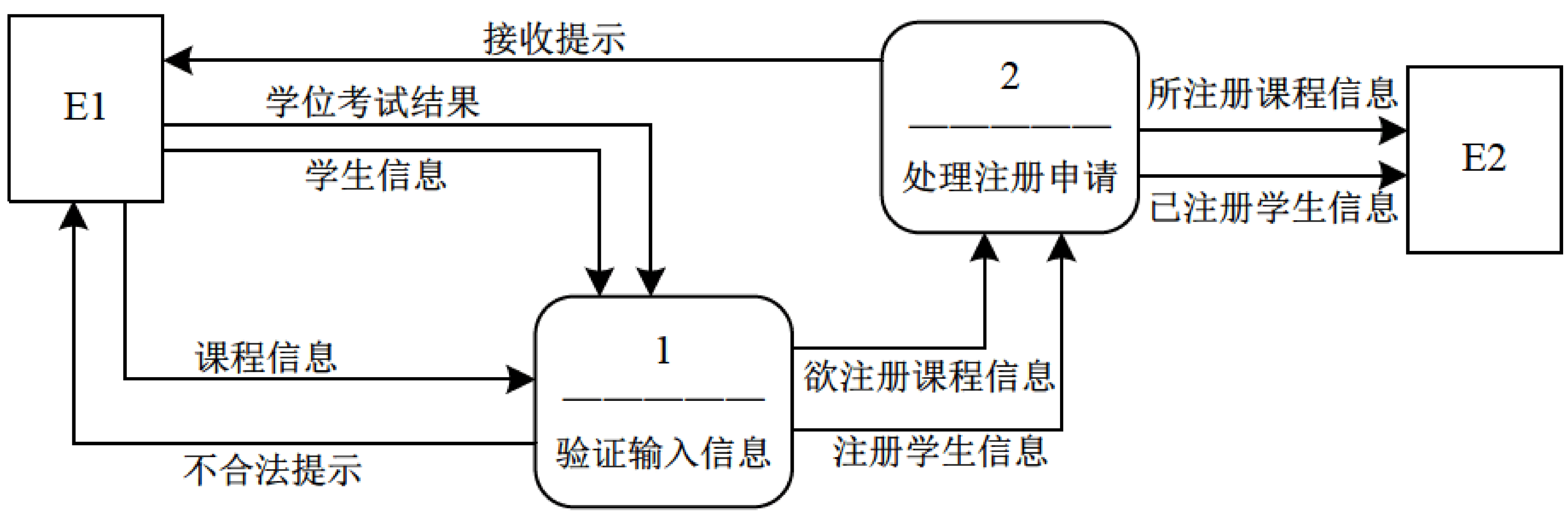


图 5-17　某课程注册系统 0 层数据流图

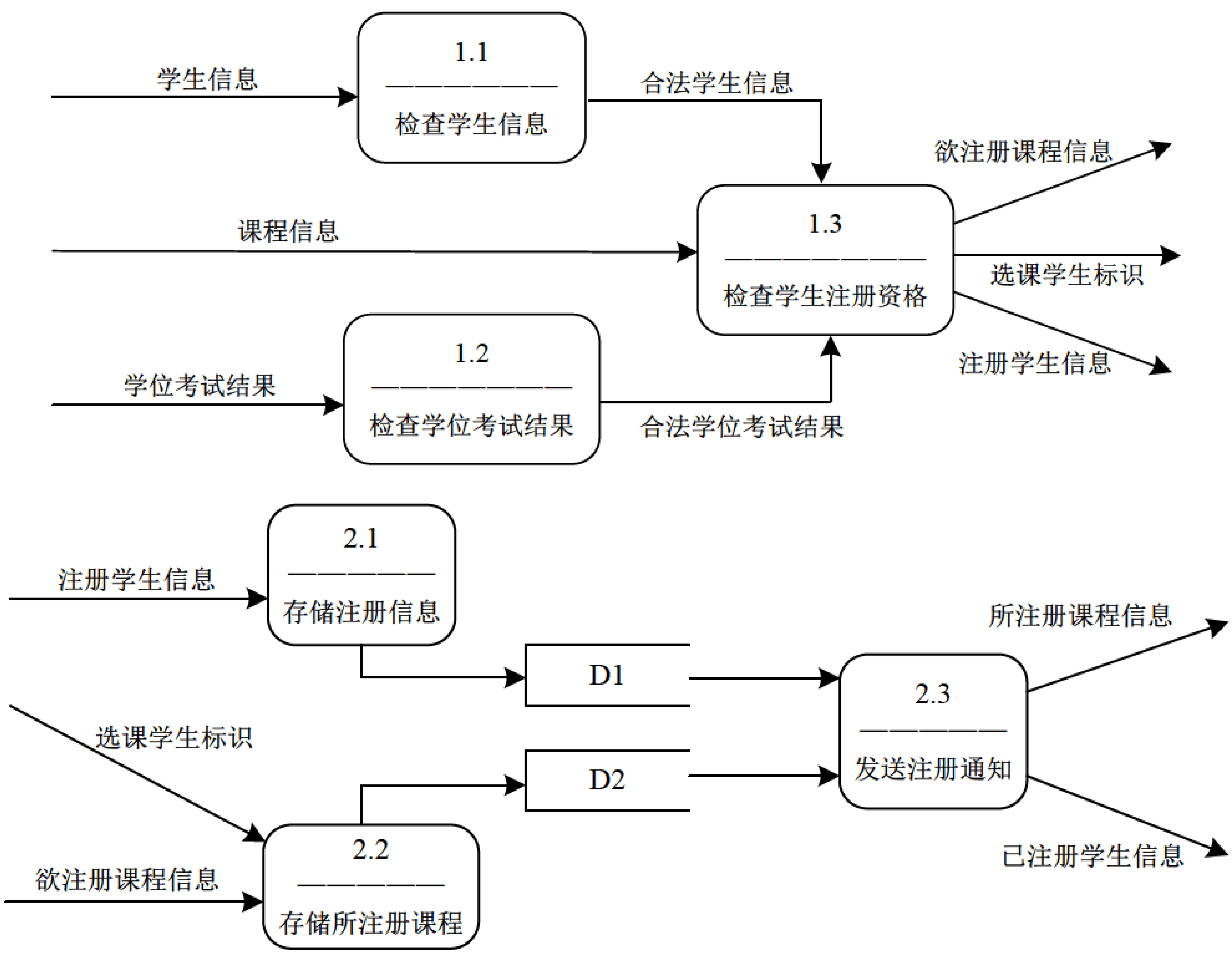


图 5-18　某课程注册系统 1 层数据流图

【问题 1】

使用说明中的词语，给出图 5-17 中的实体 E1 和 E2 的名称。

【问题 2】

使用说明中的词语，给出图 5-18 中的数据存储 D1 和 D2 的名称。

【问题 3】

根据说明和图中术语。补充图 5-18 中缺失的数据流及其起点和终点。

【问题 4】

根据补充完整的图 5-17 和图 5-18，说明上层的哪些数据流是由下层的哪些数据流组合而成。

【解析】

该题以 Web 注册系统为载体来考核考生对数据流图知识点的掌握程度。从题目的问答形式上来看，和往年相似，要求补充外部实体、补充缺失数据流、找出外部存储。解答这类问题，有两个原则：

①第一个原则是紧扣试题系统说明部分，数据流图与系统说明有着严格的对应关系，系统说明部分的每一句话都能对应到图中来，解题时可以一句一句地对照图来分析。

②第二个原则即数据的平衡原则，这一点在解题过程中也是至关重要的。数据平衡原则有两方面的含义，一方面是分层数据流图父子图之间的数据流平衡原则，另一方面是每张数据流图中输入与输出数据流的平衡原则。

【问题 1】根据 0 层数据流图 E1 向“验证输入信息”发送学位考试结果、学生信息、课程信息等，可知 E1 为学生。“处理注册申请”向 E2 发送注册课程信息和已注册信息，由此可知 E2 为教务人员。

【问题 2】根据题目中说明，存储注册信息时，“将注册学生信息记录在学生库”，可知 D1 为学生库；存储注册课程时，关联后“存入课程库”，可知 D2 为课程库。

【问题 3】对应图 5-17 中的处理 1 验证输入信息的输出数据流“不合法提示”，不难发现，在图 5-18 土中，处理 1.1 缺少了到实体学生的输出数据流“学生信息不合法提示”；处理 1.2 缺少了到实体学生的输出数据流“无注册资格提示”；处理 1.3 缺少了到实体学生的输出数据流“学位考试结果不合法提示”。再考查图 5-17 中处理 2，其输出数据流有三条，而图 5-18 中对图 5-17 中处理 2 的分解中，只包含了“所注册课程信息”和“已注册学生信息”两条数据流，缺失了“接受提示”。

【问题 4】本问题考查数据流的分解与组合。图 5-18 中对于说明中的功能出现了“学生信息不合法提示”、“无注册资格提示”和“学位考试结果不合法提示”三条数据流，说明图 5-17 中的数据流“不合法提示”是由这三条数据流组合而成。

图 5-17 中注册学生信息在图 5-18 中进一步分出注册学生信息和选课学生标识，即图 5-17 中注册学生信息是注册学生信息和选课学生标识的并集。

【答案】

【问题 1】E1：学生；E2：教务人员

【问题 2】D1：学生库；D2：课程库

【问题 3】

数据流	起点	终点
学生信息不合法提示	1.1 或 检查学生信息	E1 或 学生
无注册资格提示	1.3 或 检查学生注册资格	E1 或 学生
学位考试结果不合法提示	1.2 或 检查学位考试结果	E1 或 学生
接受提示	2.3 或 发送注册通知	E1 或 学生

【问题 4】图 5-17 中不合法提示分解为图 5-18 中的三条数据流的组合：学生信息不合法提示、无注册资格提示、学位考试结果不合法提示。

图 5-17 中注册学生信息对应图 5-18 中注册学生信息和选课学生标识。

【典型题 5-21】阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。(2012 年 11 月下午试题一)

【说明】

某电子商务系统采用以数据库为中心的集成方式改进购物车的功能，详细需求如下。

(1) 加入购物车。顾客浏览商品，点击加入购物车，根据商品标识从商品表中读取商品信息，并更新购物车表。

(2) 浏览购物车。顾客提交浏览购物车请求后，显示出购物车表中的商品信息。

(3) 提交订单。顾客点击提交订单请求，后台计算购物车表中商品的总价(包括运费)加入订单表，将购物车表中的商品状态改为待付款，显示订单详情。若商家改变价格，则刷新后可看到更改后的价格。

(4) 改变价格。商家查看订购自家商品的订单信息，根据特殊优惠条件修改价格，更新订单表中的商品价格。

(5) 付款。顾客点击付款后，系统先根据顾客表中关联的支付账户，将转账请求(验证码、价格等)提交给支付系统(如信用卡系统)进行转账；然后根据转账结果返回支付状态并更改购物车表中商品的状态。

(6) 物流跟踪。商家发货后，需按订单标识添加物流标识(物流公司、运单号)；然后可根据顾客或商家的标识以及订单标识，查询订单表中的物流标识，并从相应物流系统查询物流信息。

(7) 生成报表。根据管理员和商家设置的报表选项，从订单表、商品表以及商品分类表中读取数据，调用第三方服务 Crystal Reports 生成相关报表。

(8) 维护信息。管理员维护(增、删、改、查)顾客表、商品分类表和商品表中的信息。

现采用结构化方法实现上述需求，在系统分析阶段得到如图 5-19 所示的顶层数据流图和图 5-20 所示的 0 层数据流图。

【问题 1】

使用说明中的词语，给出图 5-19 中的实体 E1～E4 的名称。

【问题 2】

使用说明中的词语，给出图 5-20 中的数据存储 D1～D4 的名称。

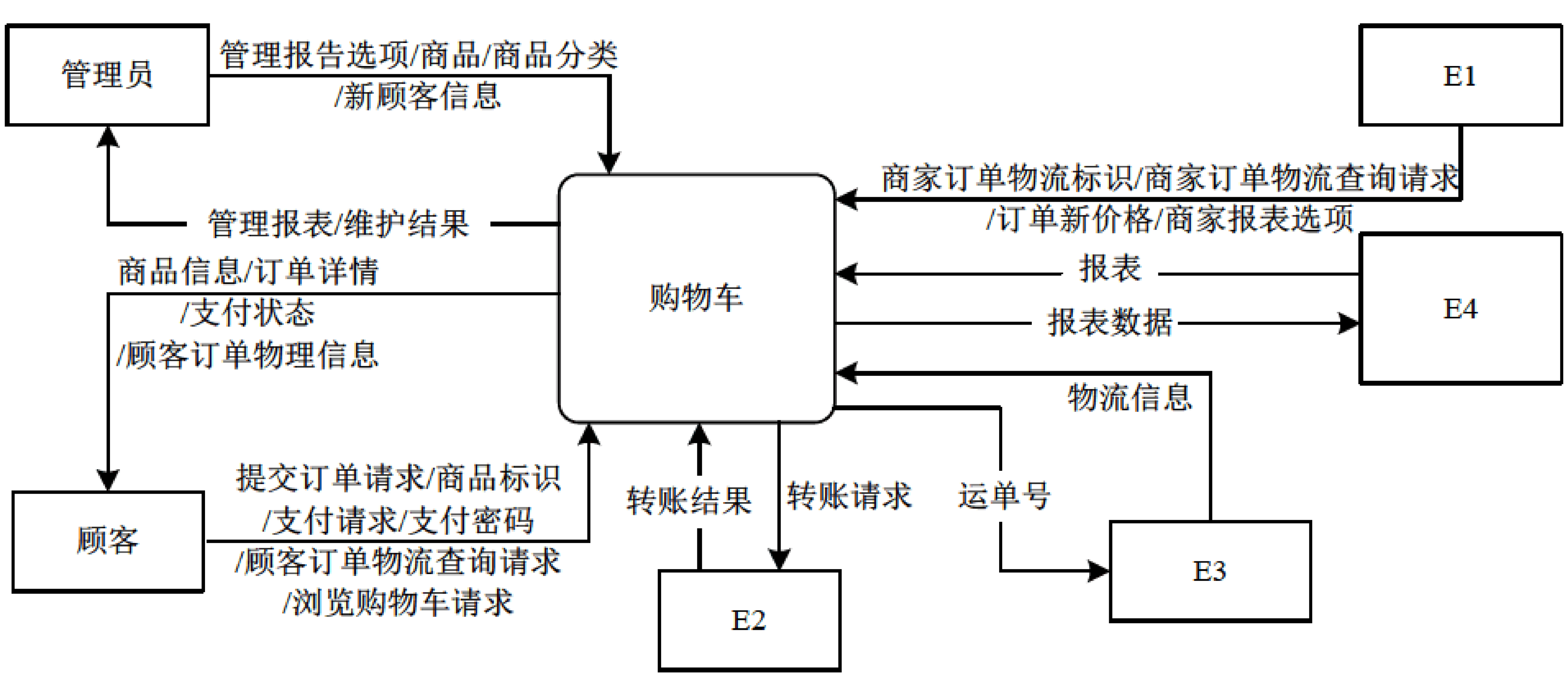


图 5-19　顶层数据流图

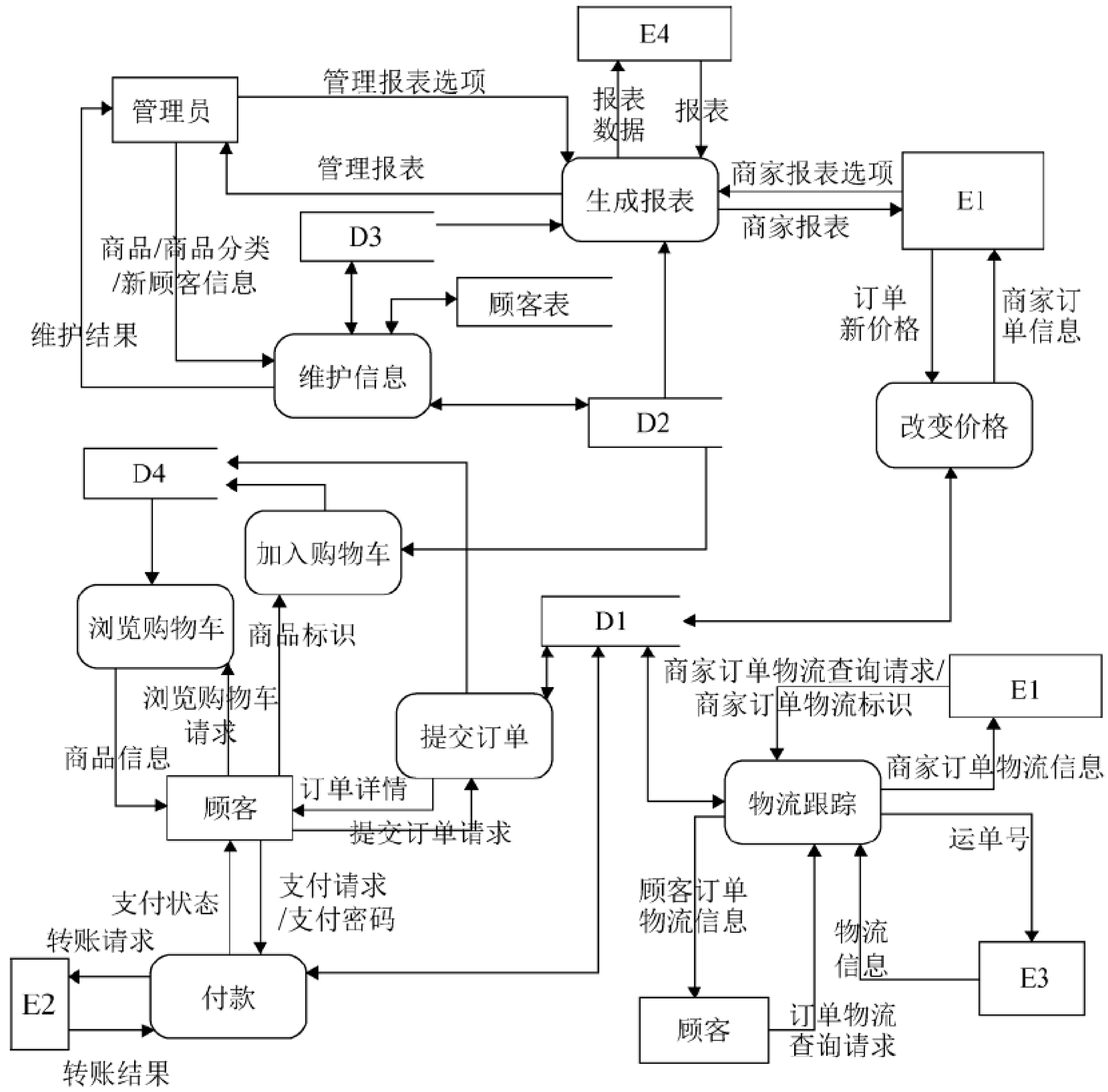


图 5-20　0 层数据流图

【问题 3】

图 5-20 中缺失了数据流，请用说明或图 5-20 中的词语，给出其起点和终点。

【问题 4】

根据说明，给出数据流“转账请求”、“顾客订单物流查询请求”和“商家订单物流查询请求”的各组成数据项。

【解析】

本题考查数据流图(DFD)的应用，采用结构化方法进行系统分析与设计，是一道传统题目，要求考生细心分析题目中所描述的内容。DFD 是一种便于用户理解、分析系统数据流程的图形化建模工具，是系统逻辑模型的重要组成部分。

【问题 1】

本问题考查顶层 DFD。顶层 DFD 一般用来确定系统边界，将待开发系统看作一个加工，因此图中只有唯一的一个处理(“购物车”)和一些外部实体，以及这两者之间的输入输出数据流。外部实体可以是使用系统的用户，也可以是为系统提供输入或接受系统输出的外部系统。本问题要求根据描述确定图中的外部实体，应仔细分析题目中的描述，并结合已经在顶层数据流图给出的数据流进行分析。

E1 到购物车的数据流是和物流相关的信息，由此定位到“物流跟踪”段，从此段的描述不难判断 E1 是商家；E2 到购物车的数据流是和转账相关的信息，由此定位到“付款”段，从此段的关键字“转账请求”和“转账结果”不难判断 E1 是支付系统；E3 到购物车的数据流的关键字为“运单号”和“物流信息”，

定位到“物流跟踪”段，从该段的最后不难判断 E3 是物流系统；E4 到购物车的数据流的关键字为“报表”，由此定位到“生成报表”段，不难判断 E4 是 Crystal Reports。

【问题 2】

本问题考查 DFD 中数据存储的确定。本题涉及的数据存储有 5 个：商品表、购物车表、订单表、顾客表和商品分类表，接下来需要结合图 5-20 所示的 0 层数据流图进行确定。D1 与“提交订单”、“改变价格”、“物流跟踪”和“付款”等加工相关，而这些加工都使用订单表，因此可以确定 D1 是订单表；然后定位到图中名为“维护信息”的加工，由“管理员维护(增、删、改、查)顾客表、商品分类表和商品表中的信息”可知，D2 和 D3 为商品分类表或商品表，D4 为购物车表。由 D2 与“加入购物车”加工的关系，可知 D2 为商品表，D3 为商品分类表。

【问题 3】

本问题考查补充分层 DFD 中的数据流。在分层 DFD 中，需要保持父图和子图的平衡，即父图中某加工的输入输出数据流必须与其子图的输入输出数据流在数量和名字上相同，或者父图的一个输入(或输出)数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是父图中的一个数据流。

由“付款”段中有关“根据转账结果返回支付状态并更改购物车表中商品的状态”的描述，存在一个起点为“付款”、终点为“D4”或“购物车表”的数据流；由“提交订单”段中有关“将购物车表中的商品状态改为待付款，显示订单详情”的描述，存在一个起点为“D4”或“购物车表”、终点为“提交订单”的数据流；由“付款”段中有关“顾客点击付款后，系统先根据顾客表中关联的支付账户，将转账请求(验证码、价格等)提交给支付系统(如信用卡系统)进行转账”的描述，存在一个起点为“顾客表”、终点为“付款”的数据流；由“生成报表”段中有关“从订单表、商品表以及商品分类表中读取数据，调用第三方服务 Crystal Reports 生成相关报表”的描述，存在一个起点为“D1”或“订单表”、终点为“生成报表”的数据流。

【问题 4】

数据流图描述了系统的分解，但没有对图中各成分进行说明。数据项是组成数据流和数据存储的最小元素。“转账请求”数据流中包含验证码、价格、账号信息等数据项；“顾客订单物流查询请求”数据流中包含顾客标识和订单标识等数据项；“商家订单物流查询请求”数据流中包含商家标识和订单标识等数据项。

【答案】

【问题 1】

E1：商家；E2：支付系统；E3：物流系统；E4：Crystal Reports

【问题 2】

D1：订单表；D2：商品表；D3：商品分类表；D4：购物车表

【问题 3】

图 5-20 中缺少的数据流：

起　点	终　点
付款	D4 或 购物车表
D4 或 购物车表	提交订单
顾客表	付款
D1 或 订单表	生成报表

【问题 4】

转账请求=验证码+价格+账号信息

顾客订单物流查询请求=顾客标识+{订单标识}

商家订单物流查询请求=商家标识+{订单标识}

【典型题 5-22】阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。(2012 年 5 月下午试题一)

【说明】

某学校开发图书管理系统，以记录图书馆所藏图书及其借出和归还情况，提供给借阅者借阅图书功能，提供给图书馆管理员管理和定期更新图书表功能。主要功能的具体描述如下。

(1) 处理借阅。借阅者要借阅图书时，系统必须对其身份(借阅者 ID)进行检查。通过与教务处维护的学生数据库、人事处维护的职工数据库中的数据进行比对，以验证借阅者 ID 是否合法，若合法，则检查借阅者在逾期未还图书表中是否有逾期未还图书，以及罚金表中的罚金是否超过限额。如果没有逾期未还图书并且罚金未超过限额，则允许借阅图书，更新图书表，并将借阅的图书存入借出图书表，借阅者归还所借图书时，先由图书馆管理员检查图书是否缺失或损坏，若是，则对借阅者处以相应罚金并存入罚金表；然后，检查所还图书是否逾期，若是，执行“处理逾期”操作；最后，更新图书表，删除借出图书表中的相应记录。

(2) 维护图书。图书馆管理员查询图书信息；在新进图书时录入图书信息，存入图书表；在图书丢失或损坏严重时，从图书表中删除该图书记录。

(3) 处理逾期。系统在每周一统计逾期未还图书，逾期未还的图书按规则计算罚金，并记入罚金表，并给有逾期未还图书的借阅者发送提醒消息。借阅者在借阅和归还图书时，若罚金超过限额，管理员收取罚金，并更新罚金表中的罚金额度。

现采用结构化方法对该图书管理系统进行分析与设计，获得如图 5-21 所示的顶层数据流图和图 5-22 所示的 0 层数据流图。

【问题 1】

使用说明中的词语，给出图 5-21 中的实体 E1～E4 的名称。

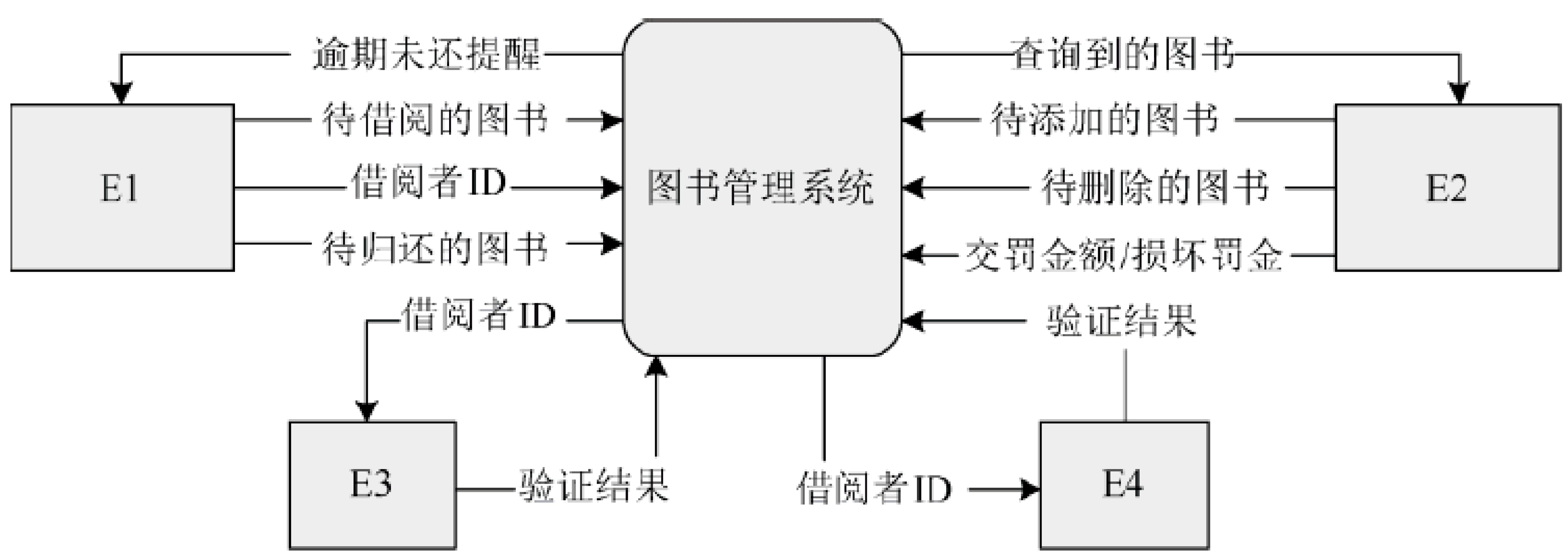


图 5-21　顶层数据流图

【问题 2】

使用说明中的词语，给出图 5-22 中的数据存储 D1～D4 的名称。

【问题 3】

在 DFD 建模时，需要对有些复杂加工(处理)进行进一步精化，绘制下层数据流图。针对图 5-21 中的加工“处理借阅”，在 1 层数据流图中应分解为哪些加工？(使用说明中的术语)

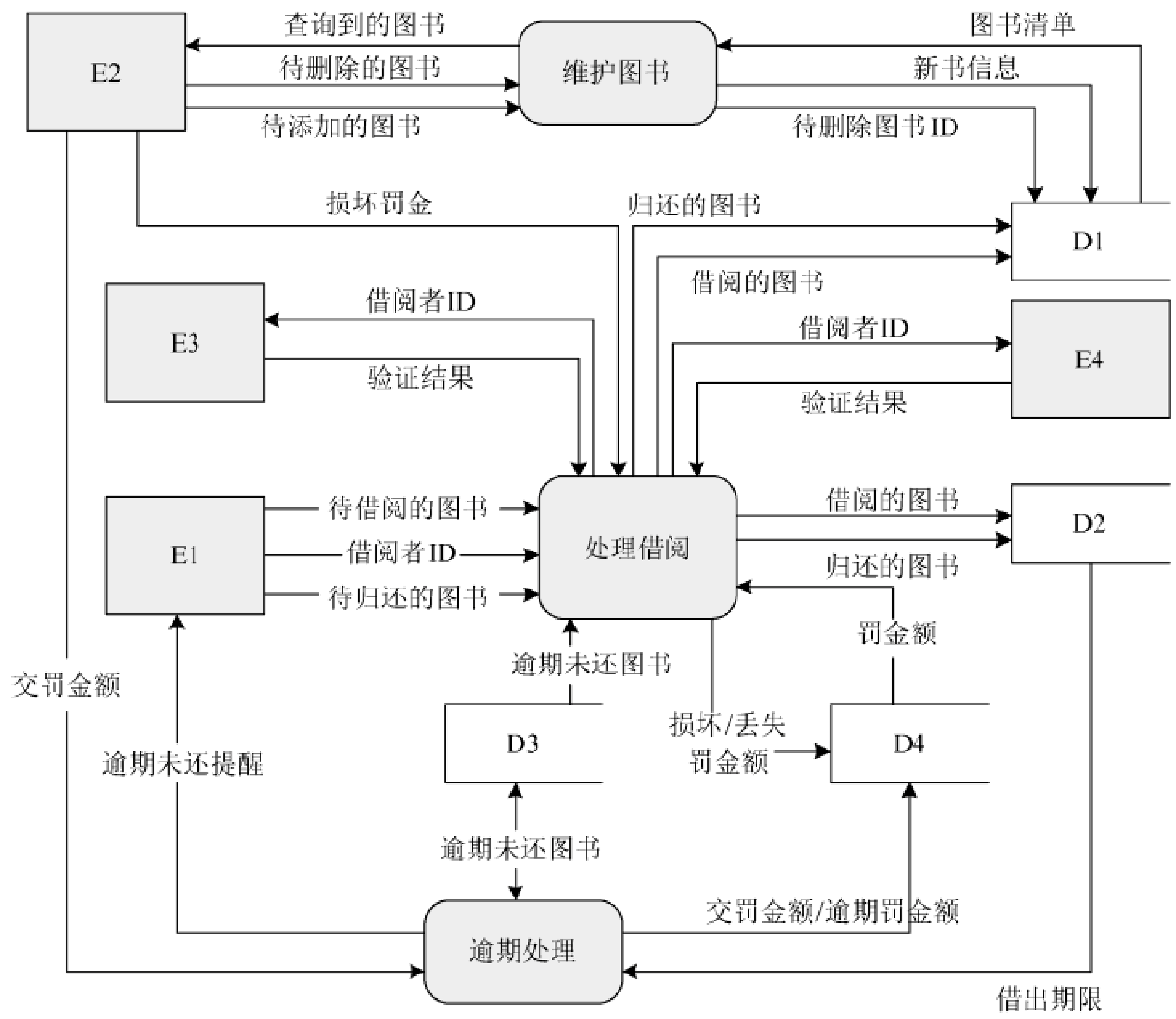


图 5-22　0 层数据流图

【问题 4】

说明【问题 3】中绘制 1 层数据流图时要注意的问题。

【解析】

【问题 1】由题目知，借阅者要借阅图书时，系统必须对其身份进行检查，而身份检查是通过验证借阅者 ID 实现的，而验证借阅者 ID 则是通过与学生数据库和职工数据库的数据进行比对实现的，E3、E4 处分别填入学生数据库和职工数据库。

图书管理系统验证借阅者的 ID 后，检查其是否有逾期未归还的图书，有的话，则给借阅者发送逾期未还提醒，很显然，E1 为借阅者。

图书维护时，要对图书进行查询、新书信息录入、书目删除等操作，进行这些操作的是图书管理员。因此 E2 为图书管理员。

【问题 2】图书管理员要从图书表中查询图书信息；新进图书时，需要将新书信息录入图书表；在图书丢失或损坏严重时，需要从图书表中删除该图书记录。由此可知，D1 为图书表。

借阅图书时，需要检查借阅者在逾期未还图书表中是否有逾期未还图书，以及罚金表中的罚金是否超过限额。D4 与罚金的操作相关，因此为罚金表。D3 与逾期处理相关，因此为逾期未还图书表。

如果系统允许借阅图书，则一方面更新图书表，另一方面将借阅的图书存入借出图书表，可见 D2 为借出图书表。

【问题 3】处理借阅涉及借书和还书两部分。借书时，首先进行借阅者身份的验证，通过验证，则在逾期未还图书表中检查是否有逾期未还图书，同时，还要在罚金表中检查罚金是否超限。还书时，需要检查图书是否缺失或损坏，然后检查图书是否逾期，最后更新图书表。

【问题 4】将加工“处理借阅”进一步分解，得到的便是 0 层数据流图的子图，原来的图便成为父图。在绘制子图的时候，要注意保持与父图的平衡。

【答案】

【问题 1】E1：借阅者；E2：图书管理员；E3/E4：学生数据库/职工数据库

【问题 2】D1：图书表；D2：借出图书表；D3：逾期未还图书表；D4：罚金表

【问题 3】检查借阅者身份或检查借阅者 ID；检查逾期未还图书；检查罚金是否超过限额；借阅图书；归还图书

【问题 4】保持父图与子图平衡。父图中某加工的输入输出流必须与它的子图的输入输出流在数量和名字上相同。如果父图的一个输入(或输出)数据流对应于子图中几个输入(或输出)数据流，而子图中组成这些数据流的数据项全体正好是父图中的这一数据流，那么它们仍然算是平衡的。

5.10　过 关 习 题

1. 模块 A 直接访问模块 B 的内部数据，则模块 A 和模块 B 的耦合类型是______。

   A. 数据耦合　　B. 标记耦合　　C. 公共耦合　　D. 内容耦合

2. 以下关于软件测试的叙述中，正确的是______。

   A. 软件测试不仅能表明软件中存在错误，也能说明软件中不存在错误
   B. 软件测试活动应从编码阶段开始
   C. 一个成功的测试能发现至今未发现的错误
   D. 在一个被测程序段中，若已发现的错误越多，则残存的错误数越少

3. 在改正当前故障的同时可能会引入新的故障，这时需要进行______。

   A. 功能测试　　B. 性能测试　　C. 回归测试　　D. 验收测试

4. 在某班级管理系统中，班级的班委有班长、副班长、学习委员和生活委员，且学生年龄在 15～25 岁。若用等价类划分来进行相关测试，则______不是好的测试用例。

   A. (队长，15)　　B. (班长，20)　　C. (班长，15)　　D. (队长，12)

5. 不属于黑盒测试技术的是______。

   A. 错误猜测　　B. 逻辑覆盖　　C. 边界值分析　　D. 等价类划分

6. 图 5-23 所示的逻辑流实现折半查找功能，最少需要______个测试用例可以覆盖所有的可能路径。

   A. 1　　B. 2　　C. 3　　D. 4

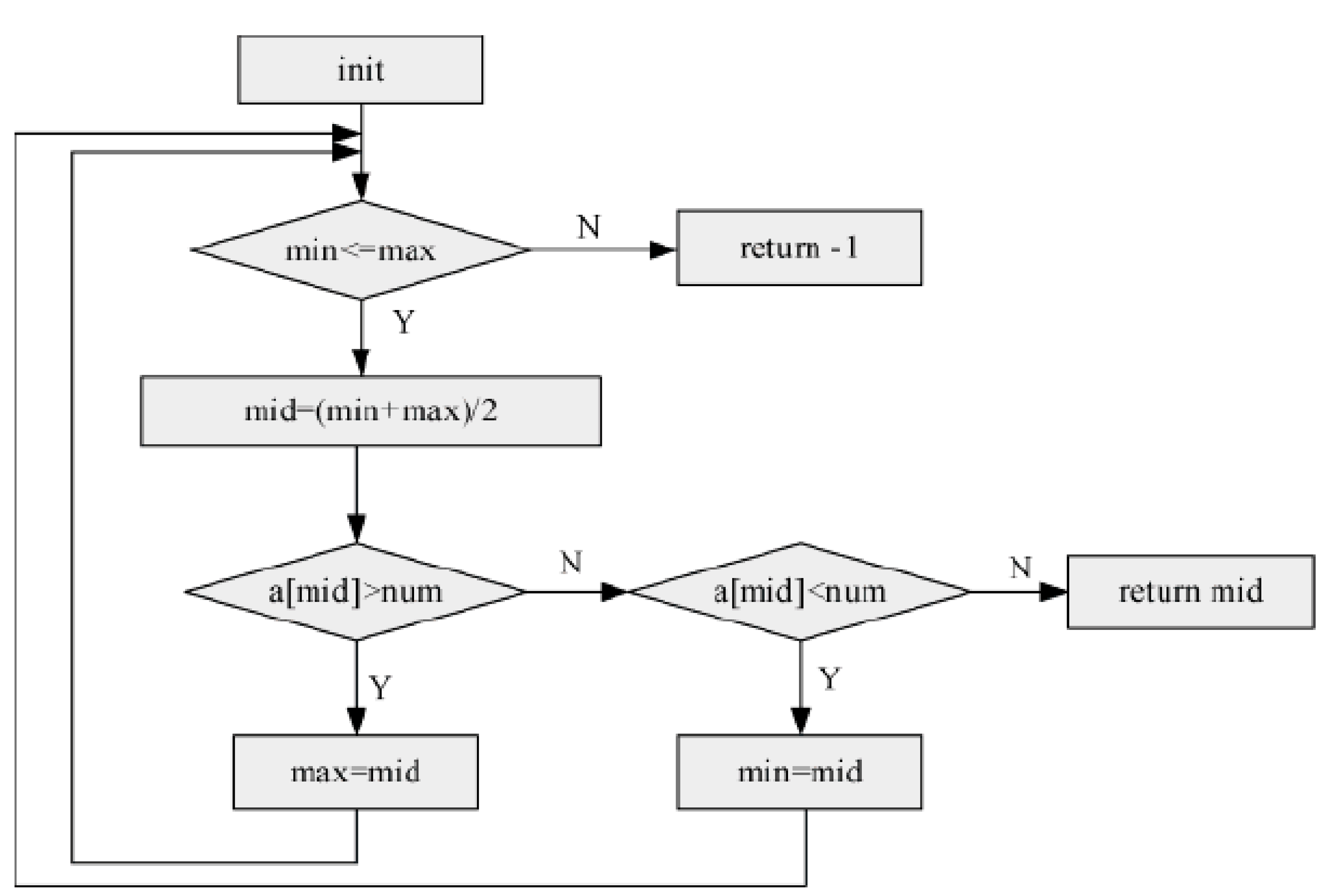


图 5-23　逻辑流

7. 图 5-24 所示的逻辑流，最少需要______个测试用例可实现语句覆盖。

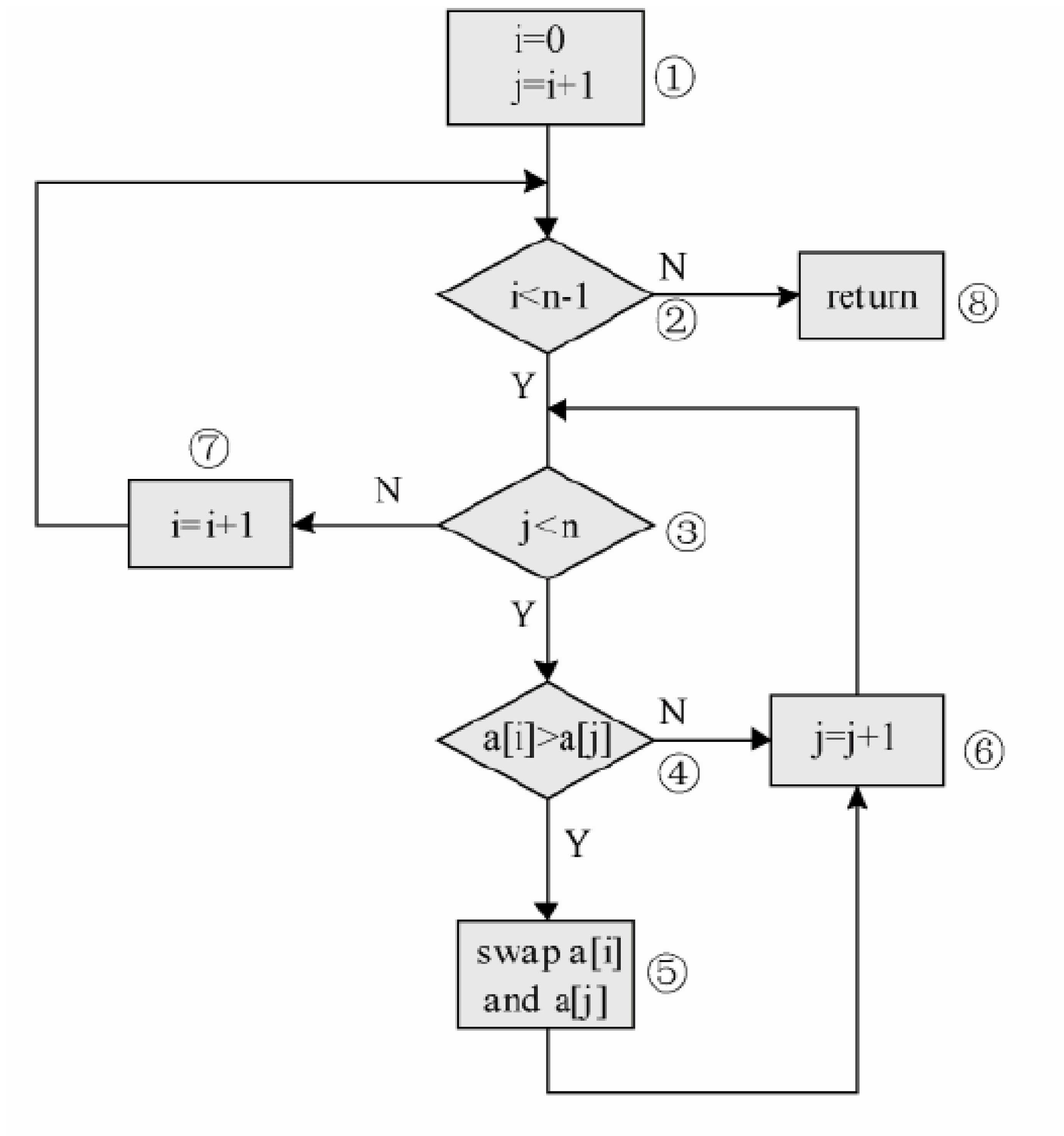


图 5-24　逻辑流

A. 1　　B. 2　　C. 3　　D. 5

8. 高质量的文档所应具有的特性中，不包括______。

A. 针对性，文档编制应该考虑读者对象群

B. 精确性，文档的行文应当十分准确，不能出现多义性的描述

C. 完整性，任何文档都应当是完整的、独立的，应该自成体系

D. 无重复性，同一软件系统的几个文档之间应该没有相同的内容，若确实存在相同内容，则可以用“见××文档××节”的方式引用

9. 以下关于软件系统文档的叙述中，错误的是______。

A. 软件系统文档既包括有一定格式要求的规范文档，又包括系统建设过程中的各种来往文件、会议纪要、会计单据等资料形成的不规范文档

B. 软件系统文档可以提高软件开发的可见度

C. 软件系统文档不能提高软件开发效率

D. 软件系统文档便于用户理解软件的功能、性能等各项指标

10. 针对应用在运行期的数据特点，修改其排序算法使其更高效，属于______维护。

A. 正确性　　B. 适应性　　C. 完善性　　D. 预防性

11. 进行防错性程序设计，可以有效地控制______维护成本。

A. 正确性　　B. 适应性　　C. 完善性　　D. 预防性

12. 软件维护阶段，为软件的运行增加监控设施属于______维护。

A. 改正性　　B. 适应性　　C. 完善性　　D. 预防性

13. 确定软件的规模划分及模块之间的调用关系是______阶段的任务。

A．需求分析　　B. 概要设计　　C. 详细设计　　D. 编码

14. 阅读下列说明和图，回答问题 1 至问题 4。

【说明】

某公司欲开发招聘系统以提高招聘效率，其主要功能如下。

(1) 接受申请。验证应聘者所提供的自身信息是否完整，是否说明了应聘职位，受理验证合格的申请，给应聘者发送致谢信息。

(2) 评估应聘者。根据部门经理设计的职位要求，审查已经受理的申请；对未被录用的应聘者进行谢绝处理，将未被录用的应聘者信息存入未录用的应聘者表，并给其发送谢绝决策；对录用的应聘者进行职位安排评价，将评价结果存入评价结果表，并给其发送录用决策，发送录用职位和录用者信息给工资系统。

现采用结构化方法对招聘系统进行分析和设计，获得如图 5-25 所示的顶层数据流图、图 5-26 所示的 0 层数据流图和图 5-27 所示的 1 层数据流图。

【问题 1】

使用说明中的术语，给出图中 E1～E3 所对应的实体名称。

【问题 2】

使用说明中的术语，给出图中 D1～D2 所对应的数据存储名称。

【问题 3】

使用说明和图中的术语，给出图 5-27 中加工 P1～P3 的名称。

【问题 4】

解释说明图 5-26 和图 5-27 是否保持平衡，若不平衡请按如下格式补充图 5-27 中数据流的名称以及数据流的起点或终点，使其平衡(使用说明中的术语或图中符号)。

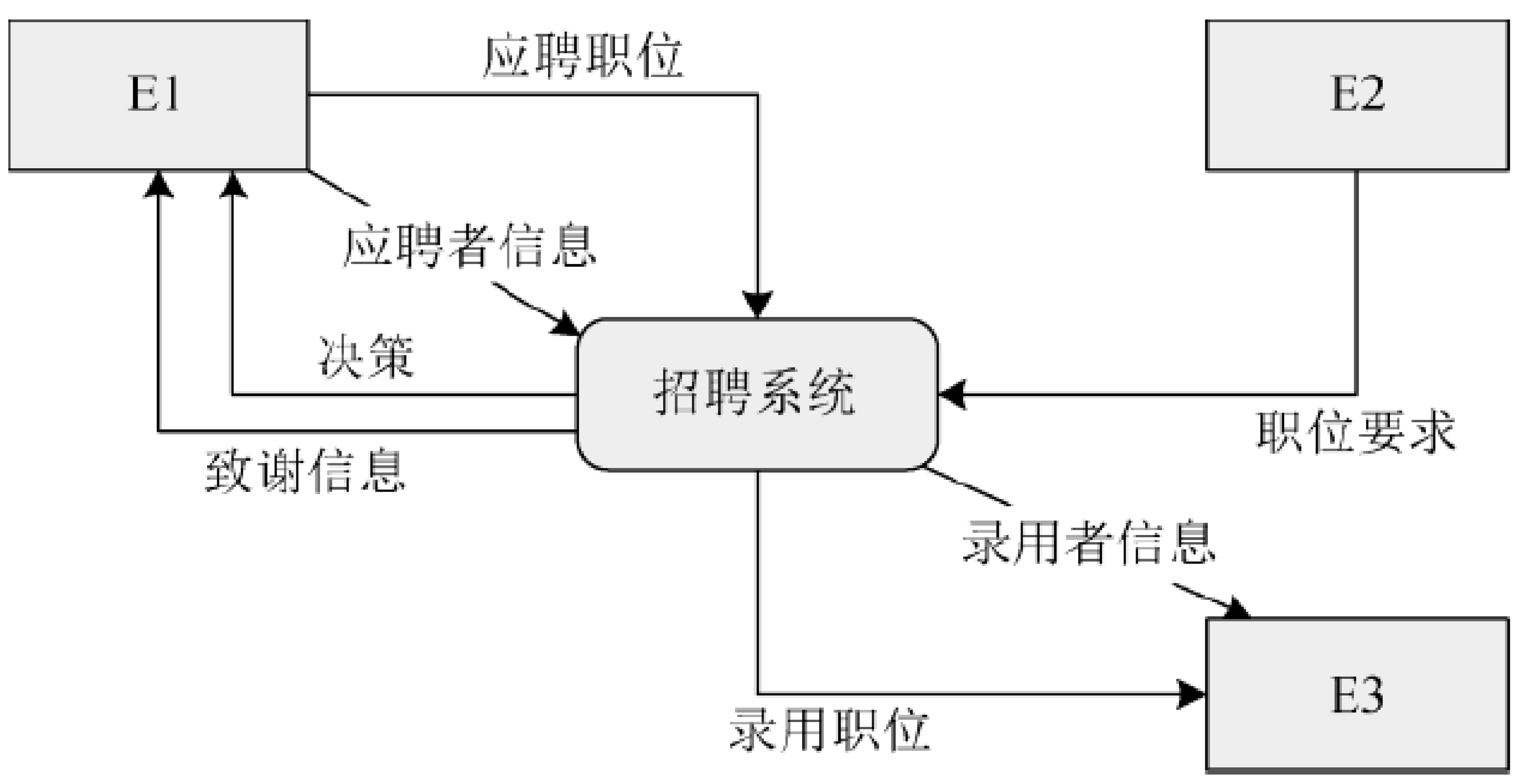


图 5-25　顶层数据流图(1)

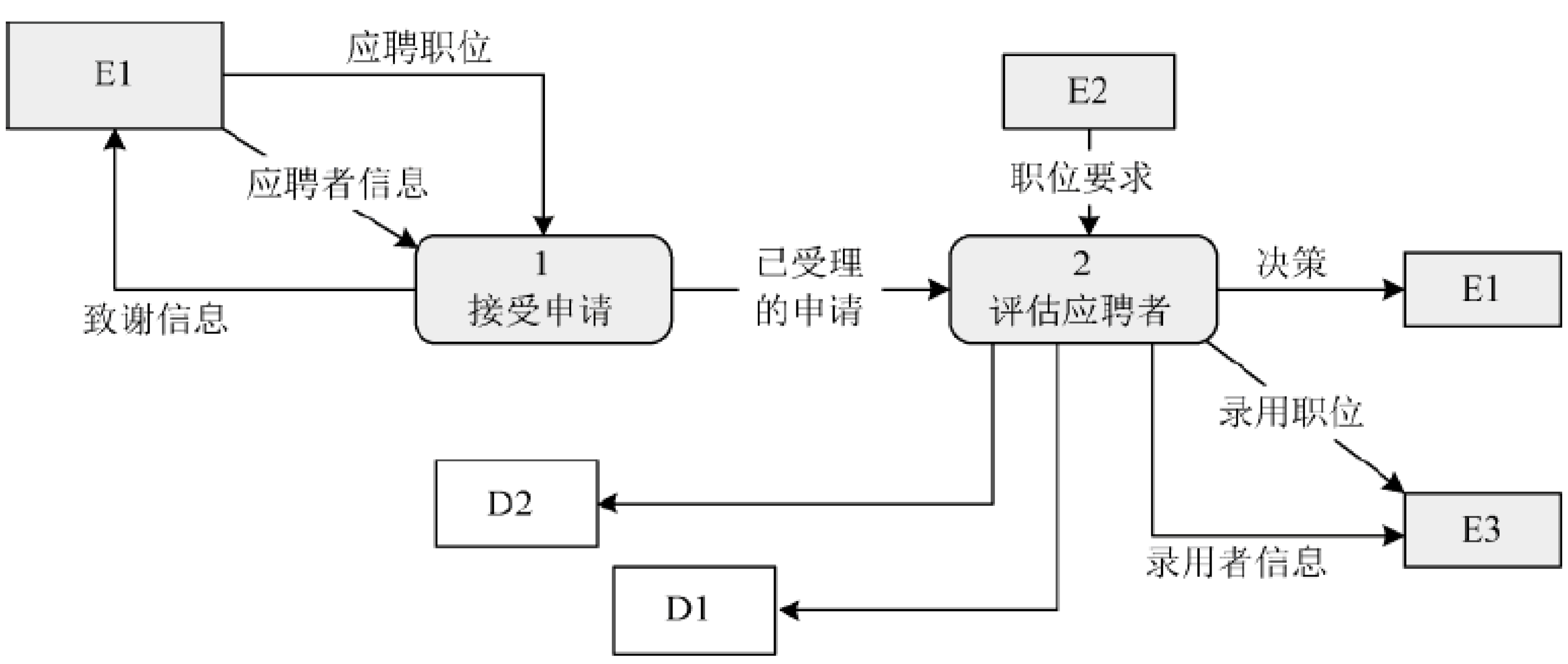


图 5-26　0 层数据流图(1)

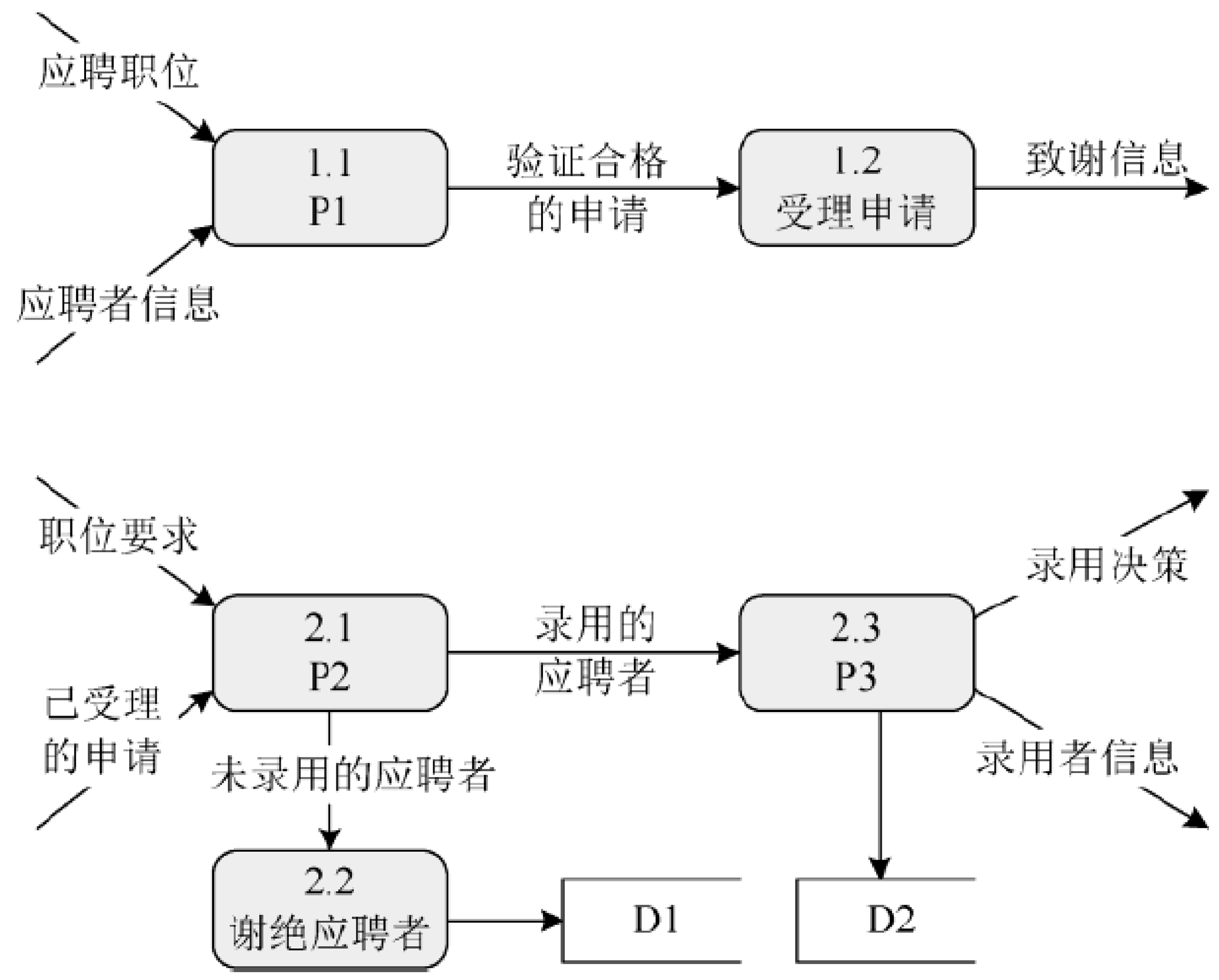


图 5-27　1 层数据流图

数据流名称	起　点

第 6 章

网络和多媒体基础知识

本章要点

- 网络体系结构
- 传输介质、传输技术、传输方法、传输控制
- 常用网络设备和各类通信设备的特点
- TCP/IP 协议簇(重点是 TCP 协议、UDP 协议、IP 协议和 ARP 协议)
- IP 地址与子网划分
- Internet 的应用
- 网络安全
- 声音信号的数字化、压缩，声音文件格式
- 图形和图像的基本概念，图像的属性、压缩编码及国际标准，图形、图像文件格式
- 动画、模拟视频和数字视频的基本概念，数字视频压缩编码，视频文件格式
- 超文本、超媒体、流媒体的概念
- 多媒体计算机系统的基本组成

6.1 网 络 概 述

考核说明：本节主要介绍计算机网络的分类和网络拓扑结构，了解即可。对于计算机网络的发展趋势，虽然没有介绍，但也要有个总体上的认识。OSI 参考模型，是计算机网络中一个很重要的概念，虽然在最近几次考试中没有考到，但仍需要了解每个层次的功能。

计算机网络是现代通信技术与计算机技术相结合的产物。计算机网络的主要功能有数据通信、资源共享、负载均衡，并且具有高可靠性。

6.1.1 计算机网络的分类

计算机网络的分类方式有很多，按照不同的分类原则，可以得到各种不同类型的计算机网络。例如，按通信距离可分为广域网、局域网和城域网；按信息交换方式可分为电路交换网、分组交换网和综合交换网；按网络拓扑结构可分为星型网、树型网、环型网和总线型网；按通信介质可分为双绞线网、同轴电缆网、光纤网和卫星网等；按传输带宽可分为基带网和宽带网；按使用范围可分为公用网和专用网；按速率可分为高速网、中速网和低速网；按通信传播方式可分为广播式和点到点式。

6.1.2 网络的拓扑结构

1. 总线型结构

总线型结构如图 6-1(a)所示，其特点为：总线型拓扑结构中只有一条双向通路，便于进行广播式传送信息；总线型拓扑结构属于分布式控制；结点的增删和位置的变动较容易；结点的接口通常采用无源电路；设备少，价格低，安装使用方便；对信号的质量要求高等。

2. 星型结构

星型结构如图 6-1(b)所示，使用中央交换处理单元以放射状连接到网中的各个结点，通常用双绞线将结点与中央单元进行连接。特点为：维护管理容易；故障隔离和检测容易；网络延迟时间短等。

3. 环型结构

环型结构如图 6-1(c)所示，信息传输线路构成一个封闭的环，各结点通过中继器连入网内，各中继器间首尾相接，信息单向沿环路逐点传送。特点为：环状网中信息的流动方向是固定的，两个结点仅有一条通路；有旁路设备；信息要串行穿过多个结点，系统响应速度慢等。

4. 树型结构

树型结构是总线结构的扩充形式，如图 6-1(d)所示，其传输介质是不封闭的分支电缆，主要用于多个网络组成的分级结构中。其特点同总线网。

5. 分布式结构

分布式结构无严格的布点规定和形状，各结点之间有多条线路相连，如图 6-1(e)所示。其特点为：有较高的可靠性；资源共享方便，网络响应时间短；结点的路由选择和流量控制难度大，管理软件复杂；硬件成本高。

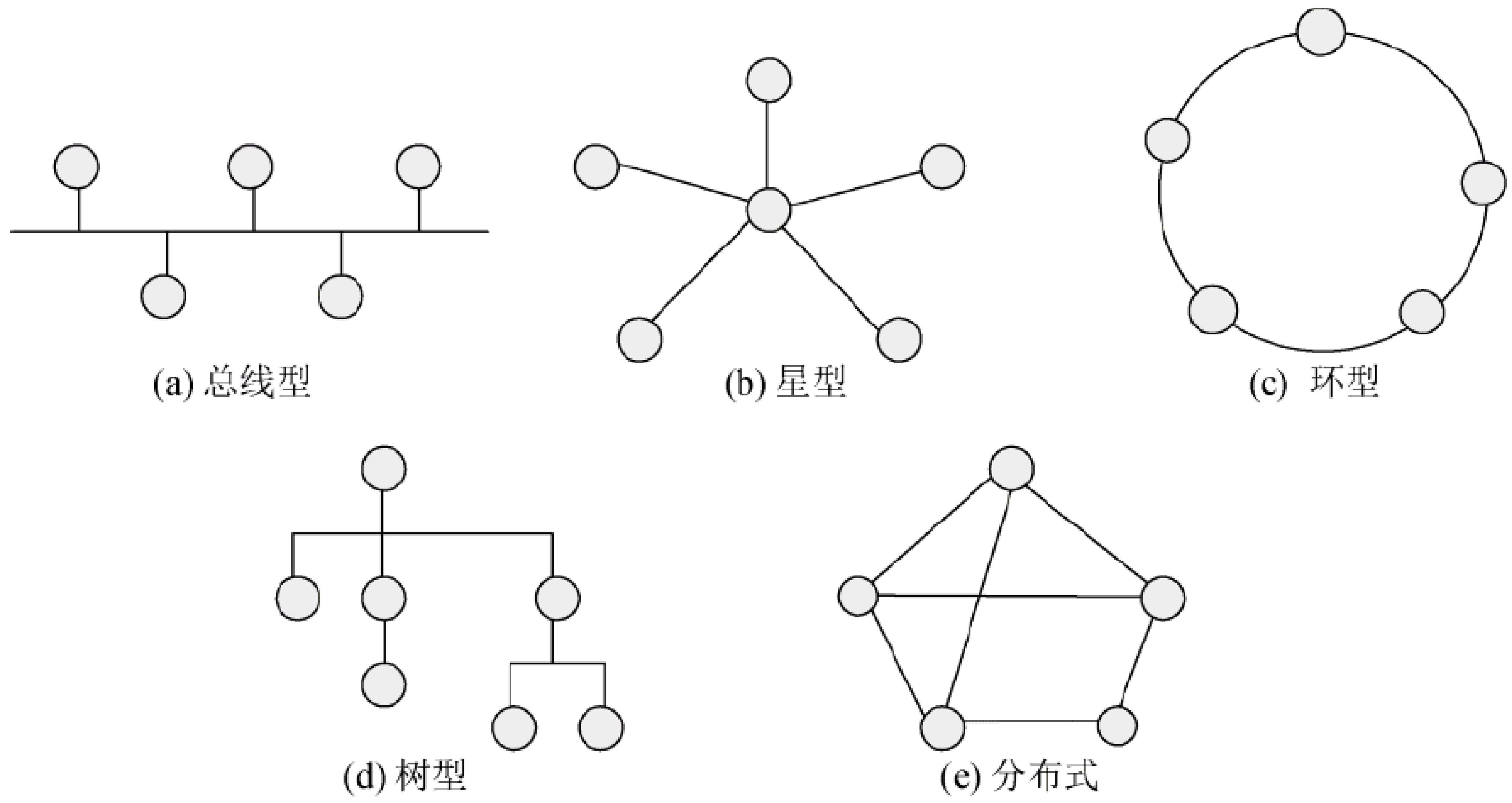


图 6-1　拓扑结构

6.1.3　ISO/OSI 网络体系结构

国际标准化组织(ISO)提出了开放系统互连参考模型(OSI)，它是一个定义异种计算机连接标准的框架结构，共有 7 层，如图 6-2 所示。

(1) 物理层：OSI 的第一层。提供为建立、维护和拆除物理链路所需的机械、电气、功能和规程的特性；提供有关在传输介质上传输非结构的位流及物理链路故障检测指示。

(2) 数据链路层：负责在两个相邻结点间的线路上无差错地传送以帧为单位的数据，并进行流量控制。

(3) 网络层：为传输层实体提供端到端的交换网络数据传送功能，使得传输层摆脱路由选择、交换方式、拥塞控制等网络传输细节；可以为传输层实体建立、维持和拆除一条或多条通信路径；对网络传输中发生的不可恢复的差错予以报错。

(4) 传输层：为会话层实体提供透明、可靠的数据传输服务，保证端到端的数据完整性；选择网络层能提供最适宜的服务；提供建立、维护和拆除传输连接功能。

(5) 会话层：为彼此合作的表示层实体提供建立、维护和结束会话连接的功能；完成通信进程的逻辑名字与物理名字间的对应；提供会话管理服务。

(6) 表示层：为应用层进程提供解释所交换信息含义的一组服务。数据的压缩、解压

缩、加密和解密等工作都由表示层负责。

(7) 应用层：提供 OSI 用户服务，即确定进程之间通信的性质，以满足用户需要以及提供网络与用户应用软件之间的接口服务。包括事务处理程序、电子邮件和网络管理程序等。

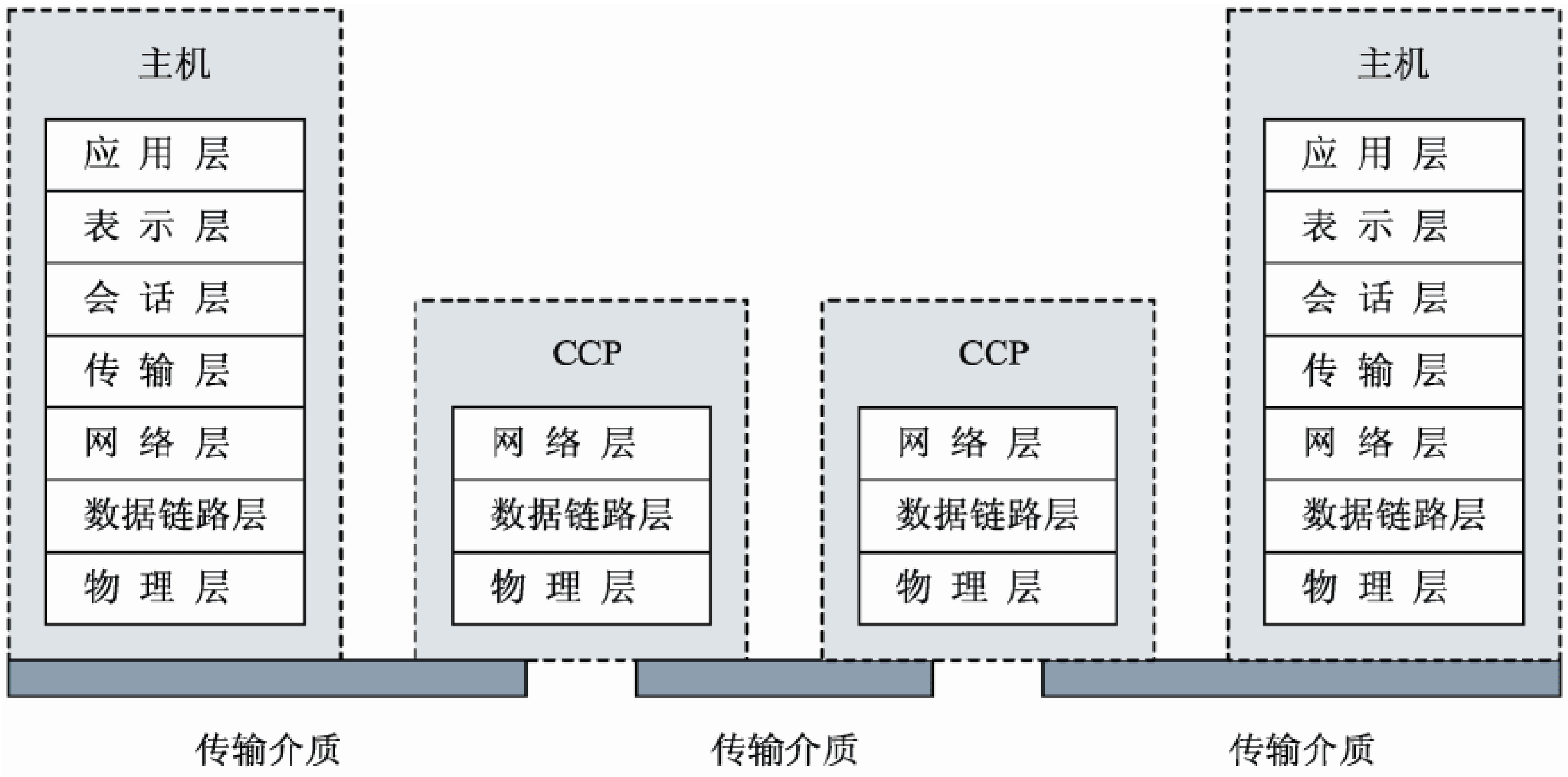


图 6-2 OSI 参考模型

6.2 网络互连硬件

考核说明： 本节主要介绍网络中的设备、传输介质和组建网络的一般方法，其中，网络设备中继器、集线器、交换机、网桥是重点，要能掌握它们的作用。

6.2.1 网络互连设备

1. 网络传输介质互连设备

网络传输介质互连设备包括 T 形头、收发器、屏蔽或非屏蔽双绞线连接器 RJ-45、RS232 接口、DB-15 接口、VB35 同步接口、网络接口单元、调制解调器等。

2. 物理层互连设备

物理层互连设备包括中继器和集线器。

(1) 中继器：由于信号在网络传输介质中有衰减和噪声，使有用的数据信号变得越来越弱，因此为了保证有用数据的完整性，并在一定范围内传送，要用中继器把接收到的弱信号分离，并再生放大以保持与原数据相同。主要优点是安装简便、使用方便、价格便宜。

(2) 集线器：可以说是一种特殊的多路中继器，具有放大信号的功能。使用双绞线的以太网多用集线器扩大网络，同时便于网络的维护。

3. 数据链路层互连设备

数据链路层互连设备包括网桥和交换机。

(1) 网桥：是一个局域网与另一个局域网之间建立连接的桥梁，它的作用是扩展网络

和通信手段，在各种传输介质中转发数据信号，扩展网络的距离，同时又有选择地将有地址的信号从一个传输介质发送到另一个传输介质，并能有效地限制两个介质系统中无关紧要的通信。

(2) 交换机：是一个具有简化、低价、高性能和高端口密集特点的交换产品。交换技术允许共享型和专用型的局域网段进行带宽调整，以减轻局域网之间信息流通出现的瓶颈问题。

4. 网络层互连设备

网络层互连设备是路由器，用于连接多个逻辑上分开的网络，具有很强的网络互连能力，具有判断网络地址和选择路径的功能。缺点是由于工作在网络层，处理的信息比网桥要多，因而处理速度比网桥慢。

5. 应用层互连设备

应用层互连设备是网关，其体现在 OSI 模型的最高层，用于转换协议，将数据重新分组，以便在两个不同类型的网络系统之间进行通信。

真题链接

【例 6-1】某单位的局域网配置如图 6-3 所示，PC2 发送到 Internet 上的报文的源 IP 地址为__(66)__。(2013 年 11 月真题 66)

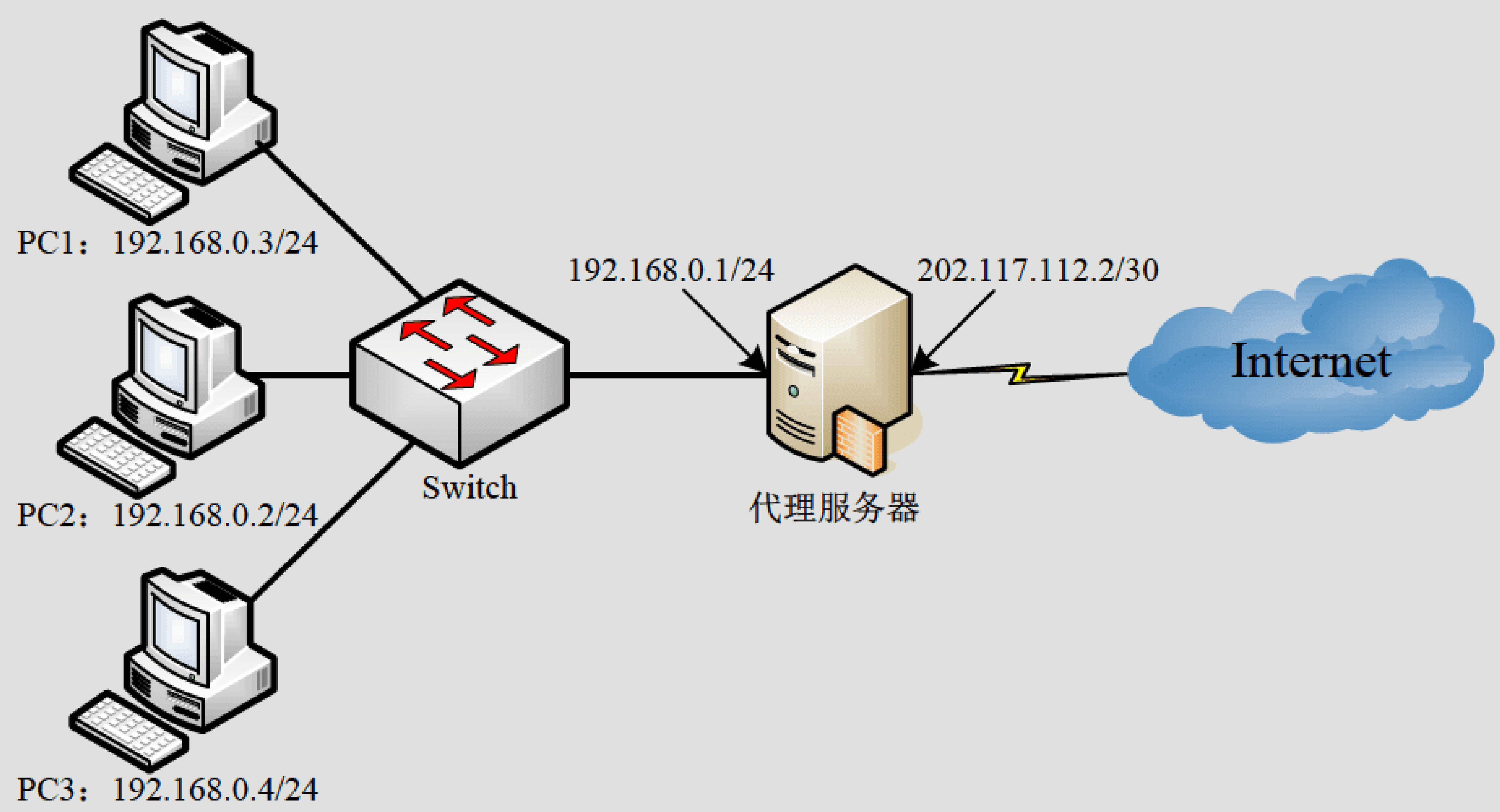


图 6-3　某单位的局域网配置图

(66) A. 192.168.0.2　　B. 192.168.0.1
C. 202.117.112.1　　D. 202.117.112.2

【解析】因为 PC1、PC2、PC3 连接的是同一个交换机，为内部局域网专用地址，所以其属于局域网，局域网和外部网进行信息交换时需要经过代理服务器的 NAT 地址转换，因此 PC2 发送到 Internet 的报文源 IP 地址会被替换为代理服务器连接外网的地址。答案为 D。

【答案】D

6.2.2 网络的传输介质

1. 有线介质

有线介质主要包括双绞线、同轴电缆、光纤等。

双绞线是现在最普通的传输介质，分为屏蔽双绞线(STP)和非屏蔽双绞线(UTP)，还可分为 3 类、4 类、5 类、6 类和 7 类双绞线，最常用的是 5 类 UTP。

同轴电缆由一对导体组成，分为基带同轴电缆和宽带同轴电缆。基带同轴电缆用来直接传输数字信号；宽带同轴电缆用于频分多路复用的模拟信号发送，还可用于不使用频分多路复用的高速数字信号发送和模拟信号发送。闭路电视所使用的 CATV 电缆就是宽带同轴电缆。

光纤利用内部全反射原理来传播光束。按光源采用的不同发光管分为发光二极管和注入型激光二极管。多模光纤使用的材料是发光二极管，单模光纤使用的是注入型二极管。

2. 无线介质

无线介质主要包括微波、红外线和激光、卫星通信。

微波的频率范围是 2～40GHz。微波通信是沿直线传播的，两微波站的通信距离一般为 30～50km，长途通信时必须建立多个中继站。微波通信分为模拟微波通信和数字微波通信，前者采用调频制，后者大都采用相移键控。

红外通信和激光通信也是沿直线传播的，需要把要传输的信号分别转换为红外光信号和激光信号，直接在空间传播。

卫星通信以人造卫星为微波中继站，它是微波通信的特殊形式。

真题链接

【例 6-2】中国自主研发的 3G 通信标准是 (70) 。(2014 年 5 月真题 70)

(70) A. CDMA2000　　B. TD-SCDMA　　C. WCDMA　　D. WiMAX

【解析】国际电信联盟(ITU)在 2000 年 5 月确定 W-CDMA、CDMA2000 和 TDS-CDMA 三大主流无线接口标准，写入 3G 技术指导性文件《2000 年国际移动通讯计划》(简称 IMT—2000)。

W-CDMA：英文名称是 Wideband Code Division Multiple Access，中文译名为宽带码分多址，它可支持 384kbps 到 2Mbps 不等的数据传输速率，支持者主要是以 GSM 系统为主的欧洲厂商。

CDMA2000：亦称 CDMA Multi-Carrier，由美国高通北美公司为主导提出，摩托罗拉、Lucent 和后来加入的韩国三星都有参与，韩国现在成为该标准的主导者。

TD-SCDMA：该标准是由中国独自制定的 3G 标准，由于中国的市场庞大，该标准受到各大主要电信设备厂商的重视，全球一半以上的设备厂商都宣布可以支持 TD-SCDMA 标准。

【答案】B

6.2.3　组建网络

1. 局域网的基本组成

在一个局域网中，其基本组成部件为服务器、客户机、网络设备、通信介质、网络软件等。

(1) 服务器：局域网的核心，可进一步分为文件服务器、打印服务器和通信服务器。

(2) 客户机：客户机又称为用户工作站，是用户与网络应用接口设备。

(3) 网络设备：主要指一些硬件设备。

(4) 通信介质：数据的传输媒体。

(5) 网络软件：主要包括底层协议软件、网络操作系统等。

2. ADSL 接入

ADSL(Asymmetric Digital Subscriber Line，非对称数字用户线)是一种使用广泛的数据传输方式。它因为上行和下行带宽不对称，因此称为非对称数字用户线。ADSL 采用频分复用技术把普通的电话线分成了电话、上行和下行三个相对独立的信道，从而避免了相互之间的干扰，即使边打电话边上网，也不会发生上网速率和通话质量下降的情况。通常 ADSL 在不影响正常电话通信的情况下可以提供很高的传输速率，其下行速率为 2～8Mbps，上行速率为 64～640kbps。

使用 ADSL 上网需要一块网卡和一个 ADSL 调制解调器。

6.3　网络的协议与标准

考核说明：本节主要介绍网络的标准、局域网协议、广域网协议和 TCP/IP 协议簇，其中，LAN 模型、以太网、TCP/IP 协议簇是考核重点。

6.3.1　网络的标准

1. 电信标准

国际电信联盟(ITU)主要由 ITU-R(无线通信部门)、ITU-T(电信标准部门)和 ITU-D(开发部门)三个部门组成。ITU-T 已经公布并使用的最重要的电信标准如下。

(1) V 系列：主要针对调制解调器的标准。

(2) X 系列：该系列标准应用于广域网，可分为如下两组。

- X.1～X.39：应用于终端形式、接口、服务设施和设备。最著名的标准是 X.25，规定了数据包装和传送的协议。
- X.40～X.199：管理网络结构、传输、发信号等。

2. 国际标准

主要的国际标准化组织如下。

- ISO——国际标准化组织。
- ANSI——美国国家标准研究所。
- NIST——美国国家标准和技术研究所。
- IEEE——电气和电子工程师协会。
- EIA——电子工业协会。

6.3.2 局域网协议

1. LAN 模型

ISO/OSI 的 7 层参考模型，本身不是一个标准，在制定具体网络协议和标准时，要以 OSI/RM 参考模型作为“参照基准”。在 IEEE 802 局域网标准中，只定义了物理层和数据链路层，并把数据链路层分成逻辑链路控制(LLC)子层和介质访问控制(MAC)子层，如图 6-4 所示。

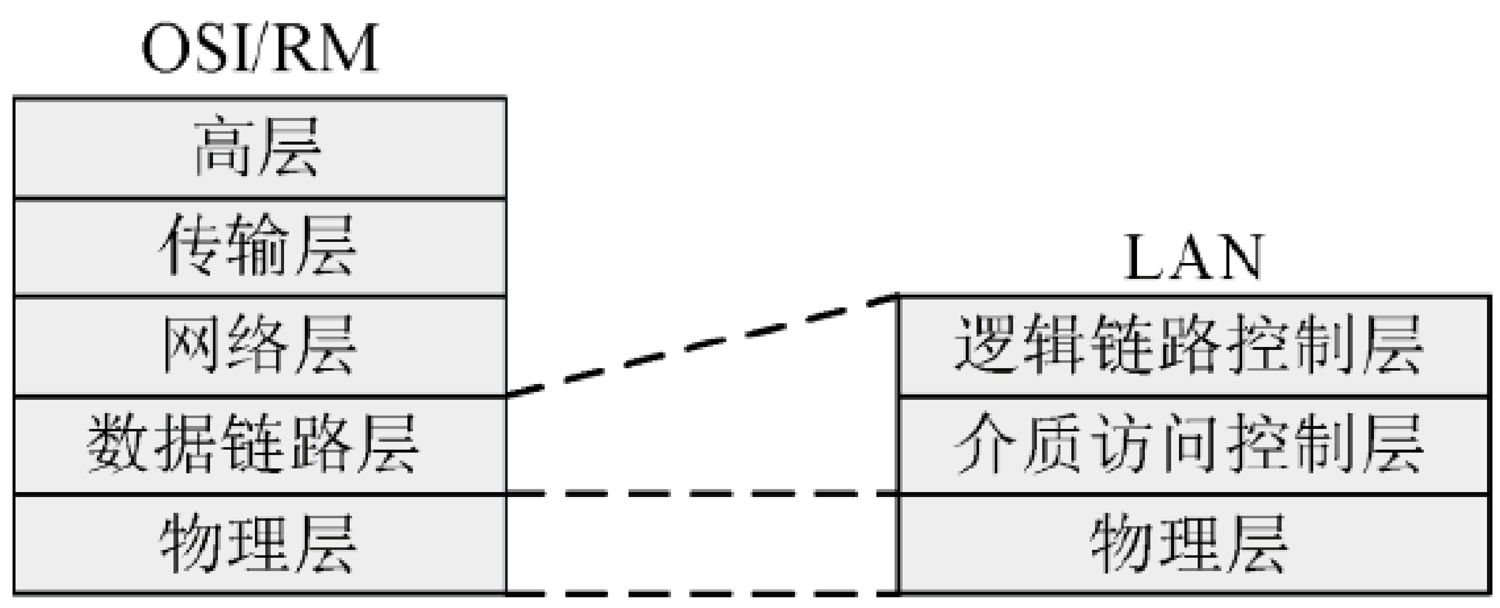


图 6-4　LAN 模型的层次

(1) 物理层：主要处理在物理链路上发送、传递和接收的非结构化的比特流。

(2) MAC 层：控制对传输介质的访问和对信道资源的分配，实现帧的寻址和识别，完成帧检测序列和检验等功能。

(3) LLC 层：提供可靠的信道、数据帧的封装和拆除，为高层提供网络服务的逻辑接口，能够实现差错控制和流量控制。

2. 以太网(IEEE 802.3 标准)

目前以太网主要包括三种类型。

- IEEE 802.3 中定义的标准局域网，速度为 10Mbps，传输介质为细同轴电缆。
- IEEE 802.3u 中定义的快速以太网，速度为 100Mbps，传输介质为双绞线。
- IEEE 802.3z 中定义的千兆以太网，速度为 1000Mbps，传输介质为光纤或双绞线。

3. 令牌环网(IEEE 802.5)

令牌环是环状网中最普遍采用的介质访问控制，它是用于环状网结构的分布式介质访问控制，其流行性仅次于以太网。令牌环网的传输介质主要基于屏蔽双绞线、非屏蔽双绞线两种；拓扑结构可以有多种，如环状、星状、总线；编码方法为差分曼彻斯特编码。

4. FDDI(光纤分布式数据接口)

类似令牌环网的协议，用光纤作为传输介质，数据传输速率可达 100Mbps，环路长度可扩展到 200km，连接的站点数可以达到 1000 个。它采用双环体系结构，两环上的信息反方向流动。双环中的一环称为主环，另一环称为次环。

6.3.3　广域网协议

1. 点对点协议

点对点协议(PPP)主要通过拨号或专线方式建立点对点连接发送数据，使其成为各种主机、网桥和路由器之间简单连接的一种共通的解决方案。优点在于简单、具备用户验证能力、可以解决 IP 分配等。

2. 数字用户线

xDSL 是各种数字用户线的统称，主要 DSL 技术和产品有以下几种。

- ADSL——不对称数字用户线。
- SDSL——单对线数字用户环路。
- IDSL——ISDN 用的数字用户线。
- RADSL——速率自适应非对称型数字用户线。
- VDSL——甚高速数字用户线。

3. 数字专线

数字数据网(Digital Data Network，DDN)是采用数字传输信道传输数据信号的通信网，可提供点对点、点对多点透明传输的数据专线出租线路，为用户传输数据、图像、声音等信息。数字数据网是以光纤为中继干线网络，以结点为基本单位(结点间通过光纤连接，构成网状的拓扑结构)，采用固定连接的方式，直接进入电信的 DDN 网络。

4. 帧中继

帧中继(FR)是在用户网络接口之间提供用户信息流的双向传送，并保持顺序不变的一种承载业务。用户信息以帧为单位进行传输，并对用户信息流进行统计复用。帧中继提供一种简单的面向连接的虚电路分组服务，包括交换虚电路连接和永久虚电路连接。

5. 异步传输模式

异步传输模式(ATM)是一种面向分组的快速分组交换模式，使用了异步时分复用技术，将信息流分割成固定长度的信元。信元由信元头和信元体构成，信元头占 5 个字节，信元体占 48 个字节。

ATM 参考模型由四层构成。

- 用户层：由用户平面、控制平面和管理平面组成。
- ATM 适配层：负责将用户层的信息转换成 ATM 网络可用的格式。
- ATM 层：负责生成信元。

- 物理层：负责对信元进行编码，并将其交给物理介质。

6. X.25 协议

X.25 协议在本地和远程之间提供一个全双工、同步的透明信道，并定义了三个相互独立的控制层：物理层、链路层和分组层，它们分别对应于 ISO/OSI 的物理层、链路层和网络层。X.25 是在公共数据网上，以分组方式进行操作的 DTE 和 DCE 之间的接口。

6.3.4 TCP/IP 协议簇

TCP/IP 作为 Internet 的核心协议，被广泛应用于局域网和广域网中，目前已成为事实上的国际标准。TCP/IP 包含许多重要的基本特性，这些特性主要表现在五个方面：逻辑编址、路由选择、域名解析、错误检测与流量控制以及对应用程序的支持等。

1. TCP/IP 分层模型

TCP/IP 是 Internet 的基础和核心，和 OSI 参考模型一样，也是采用层次体系结构，从上至下分为应用层、传输层、网际层和网络接口层。

(1) 应用层：处在分层模型的最高层，用户调用应用程序来访问 TCP/IP 互联网络，以享受网络上提供的各种服务。

(2) 传输层：提供应用程序之间的通信服务。这种通信又叫端到端的通信。传输层既要系统地管理数据信息的流动，还要提供可靠的传输服务，以确保数据准确而有序地到达目的地。

(3) 网际层：又称为 IP 层，主要处理机器之间的通信问题。它接受传输层请求，传送某个具有目的地址信息的分组。该层主要完成：把分组封装到 IP 数据报中，填入数据报的首部，使用路由算法选择；处理接收到的数据报；适时发出 ICMP 的差错和控制报文，并处理收到的 ICMP 报文。

(4) 网络接口层：处在 TCP/IP 的最底层，主要负责管理为物理网络准备数据所需的全部服务程序和功能。该层包含设备驱动程序，也可能是一个复杂的使用自己的数据链路协议的子系统。

2. 网络接口层协议

TCP/IP 不包含具体的物理层和数据链路层，只定义了网络接口层作为物理层与网络层的接口规范。这个物理层可以是广域网，如 X.25 公用数据网；可以是局域网，如 Ethernet、Token-Ring 和 FDDI 等。任何物理网络只要按照这个接口规范开发网络接口驱动程序，都能够与 TCP/IP 集成。网络接口层处在 TCP/IP 的最底层，主要负责管理为物理网络准备数据所需的全部服务程序和功能。

3. 网际层协议——IP

IP 所提供的服务通常被认为是无连接的和不可靠的。事实上，在网络性能良好的情况下，IP 传送的数据能够完好无损地到达目的地。

IP 的主要功能包括：将上层数据(如 TCP、UDP 数据)或同层的其他数据(如 ICMP 数据)

封装到 IP 数据报中；将 IP 数据报传送到最终目的地；为了使数据能够在链路层上进行传输，对数据进行分段；确定数据报到达其他网络中的目的地的路径。

4. ARP 和 RARP

地址解析协议(Address Resolution Protocol，ARP)及反地址解析协议(RARP)是驻留在网际层中的另一个重要协议。ARP 的作用是将 IP 地址转换为物理地址，RARP 的作用是将物理地址转换为 IP 地址。

5. 网际层协议——ICMP

Internet 控制信息协议(Internet Control Message Protocol，ICMP)是网际层的另一个比较重要的协议。由于 IP 是一种尽力传送的通信协议，即传送的数据报可能丢失、重复、延迟或乱序，因此 IP 需要一种避免差错并在发生差错时报告的机制。ICMP 就是一个专门用于发送差错报文的协议。ICMP 定义了五种差错报文(源抑制、超时、目的不可达、重定向和要求分段)和四种信息报文(回应请求、回应应答、地址屏蔽码请求和地址屏蔽码应答)。IP 在需要发送一个差错报文时要使用 ICMP，而 ICMP 也是利用 IP 来传送报文的。

6. 传输层协议——TCP

TCP(Transmission Control Protocol，传输控制协议)是整个 TCP/IP 协议簇中最重要的协议之一。它在 IP 提供的不可靠数据服务的基础上，为应用程序提供了一个可靠的、面向连接的、全双工的数据传输服务。

TCP 采用了一个叫重发(Retransmission)的技术实现可靠传输。具体来说，在 TCP 传输过程中，发送方启动一个定时器，然后将数据包发出，当接收方收到这个信息时就给发送方一个确认(Acknowledgement)信息。而如果发送方在定时器到点之前没收到这个确认信息，就重新发送这个数据包。

7. 传输层协议——UDP

用户数据报协议(User Datagram Protocol，UDP)是一种不可靠的、无连接的协议，可以保证应用程序进程间的通信。与同样处在传输层的面向连接的 TCP 相比较，UDP 是一种无连接的协议，它的错误检测功能要弱得多。可以这样说：TCP 有助于提供可靠性；而 UDP 则有助于提高传输的高速率性。

8. 应用层协议

随着计算机网络的广泛应用，人们已经有了许多基本的、相同的应用需求。为了让不同平台的计算机能够通过计算机网络获得一些基本的、相同的服务，也就应运而生了一系列应用级的标准，实现这些应用标准的专用协议被称为应用级协议，相对于 OSI 参考模型来说，它们处于较高的层次结构，所以也称为高层协议。应用层的协议有 NFS、Telnet、SMTP、HTTP、DNS、SNMP 和 FTP 等。

真题链接

【例 6-3】PPP 中的安全认证协议是 (66) ，它使用三次握手的会话过程传送密文。(2014 年 11 月真

题 66)

(66) A. MDS　　B. PAP　　C. CHAP　　D. NCP

【解析】一、安全认证介绍

1. PPP 的 NCP 可以承载多种协议的三层数据包。

2. PPP 使用 LCP 控制多种链路的参数(建立、认证、压缩、回拨)。

二、PPP 的认证类型

1. PPP 的 pap 认证是通过二次握手建立认证(明文不加密)。

2. PPP 的 chap 挑战握手认证协议，通过三次握手建立认证(密文采用 MDS 加密)。

3. PPP 的双向验证，采用的是 chap 的主验证风格。

4. PPP 的加固验证，采用的是两种(pap,chap)验证同时使用。

【答案】C

【例 6-4】ICMP 协议属于因特网中的<u>(67)</u>协议，ICMP 协议数据单元封装在<u>(68)</u>中传送。(2014 年 11 月真题 67～68)

(67) A. 数据链路层　　B. 网络层　　C. 传输层　　D. 会话层

(68) A. 以太帧　　B. TCP 段　　C. UDP 数据报　　D. IP 数据报

【解析】ICMP 是 TCP/IP 协议簇的一个子协议，属于网络层协议，主要用于在主机与路由器之间传递控制信息，包括报告错误、交换受限控制和状态信息等。和 TCP、UDP、IGMP 等协议一样，ICMP 协议使用 IP 数据报传送数据。

【答案】(67)B；(68)D

6.4　Internet 及其应用

考核说明：本节主要介绍 Internet 地址和 Internet 提供的服务，需要重点掌握 IP 地址的分类和子网划分，这是必考内容。同时，也需要了解 Internet 提供的各种服务。

从用户的角度来看，整个 Internet 在逻辑上是统一的、独立的，在物理上则由不同的网络互联而成。从技术角度看，Internet 本身不是某一种具体的物理网络技术，它是能够互相传递信息的众多网络的一个统称，或者说它是一个网间网，只要人们进入这个因特网，就是在使用 Internet。

6.4.1　Internet 地址

1. 域名

域名(Domain Name)通常是用户所在的主机名字或地址。域名格式由若干部分组成，每个部分又称子域名，它们之间用“.”分隔，每个部分最少由两个字母或数字组成。域名通常按分层结构来构造，每个子域名都有其特定的含义。通常情况，一个完整、通用的层次型主机域名由如下四部分组成。

计算机主机名.本地名.组名.最高层域名

2. IP 地址

Internet 地址是按名字来描述的，这种地址表示方式易于理解和记忆。实际上，Internet 中的主机地址是用 IP 地址来唯一标识的。

IP 地址的长度为 32 位，分为 4 段，每段 8 位，可以用十进制数和二进制数表示。二进制格式是计算机能识别的格式，十进制格式是由二进制格式“翻译”过来的。每段数字的范围为 0～255，段与段之间用句点分隔。IP 地址由两部分组成，一部分为网络地址，另一部分为主机地址。

IP 地址分为 A、B、C、D、E 共五类。

(1) A 类 IP 地址。由 1 个字节的网络地址和 3 个字节的主机地址组成，网络地址的最高位必须是 0，地址范围是 1.0.0.1～126.255.255.254。可用的 A 类网络有 126 个，每个网络能容纳 1677214 个主机。

(2) B 类 IP 地址。由 2 个字节的网络地址和 2 个字节的主机地址组成，网络地址的最高位必须是 10，地址范围是 128.10.1～191.255.255.254。可用的 B 类网络有 16384 个，每个网络能容纳 65534 个主机。

(3) C 类 IP 地址。由 3 个字节的网络地址和 1 个字节的主机地址组成，网络地址的最高位必须是 110，地址范围是 192.0.1.1～223.255.255.254。C 类网络可达 2097152 个，每个网络能容纳 254 个主机。

(4) D 类 IP 地址。用于多点广播(Multicast)。第一个字节以 1110 开始，它是一个专门保留的地址，并不指向特定的网络，目前这一类地址被用在多点广播中。多点广播地址用来一次寻址一组计算机，它标识共享同一协议的一组计算机。地址范围是 224.0.0.1～239.255.255.254。

(5) E 类 IP 地址。以 1111 开始，为将来使用保留，仅做实验和开发用。

另外，全零(0.0.0.0)地址指任意网络。全“1”的 IP 地址(255.255.255.255)是当前子网的广播地址。

网络软件和路由器使用子网掩码来识别报文是仅存放在网络内部还是被路由转发到其他地方。在一个字段内，1 的出现表明一个字段包含所有或部分网络地址，0 表明主机地址位置。如 C 类地址的子网掩码为 255.255.255.0。

3. NAT 技术

因特网面临 IP 地址短缺的问题。解决这个问题有所谓长期的和短期的两种解决方案。长期的解决方案就是使用具有更大地址空间的 IPv6 协议，网络地址翻译(Network Address Translators，NAT)是许多短期的解决方案中的一种。NAT 的实现主要有两种形式：动态地址翻译(Dynamic Address Translation)和 m：1 翻译(这种技术也叫作伪装)。

4. IPv6 简介

IPv6 具有长达 128 位的地址空间，可以彻底解决 IPv4 地址不足的问题。

1) IPv6 数据包的格式

IPv6 数据包有一个 40B 的基本首部(Base Header)，其后可允许有 0 个或多个扩展首部(Extension Header)，再后面是数据。每个 IPv6 数据包都是从基本首部开始。

2) IPv6 的地址表示

一般来讲，一个 IPv6 数据包的目的地址可以是以下三种基本类型地址之一。

- 单播(Unicast)：传统的点对点通信。
- 多播(Multicast)：一点对多点的通信，数据包交付到一组计算机中的每一个。IPv6 没有采用广播的术语，而是将广播看作多播的一个特例。
- 任播(Anycast)：这是 IPv6 增加的一种类型。任播的目的站是一组计算机，但数据包在交付时只交付给其中的一个，通常是距离最近的一个。

为了使地址表示简洁，IPv6 使用冒号十六进制记法，它把每个 16 位用相应的十六进制表示，各组之间用冒号分隔，如：

235A:3C78:FFF0:0:73B:0:0:B5

冒号十六进制记法允许 0 压缩，即一连串连续的 0 可以用一对冒号来取代，但 0 压缩只能用一次。例如，上述的 IPv6 地址可以压缩为

235A:3C78:FFF0:0:73B:0: :B5

真题链接

【例 6-5】分配给某公司网络的地址块是 210.115.192.0/20，该网络可以被划分为<u>(70)</u>个 C 类子网。(2014 年 11 月真题 70)

(70) A. 4　　B. 8　　C. 16　　D. 32

【解析】子网掩码的前 20 位为 1，还有 4 位可以表示不同的子网，故最多可以划分为 2^4=16 个子网。

【答案】C

【例 6-6】IP 地址块 155.32.80.192/26 包含了<u>(66)</u>个主机地址，以下 IP 地址中，不属于这个网络的地址是<u>(67)</u>。(2014 年 5 月真题 66～67)

(66) A. 15　　B. 32　　C.62　　D.64

(67) A. 155.32.80.202　　B.155.32.80.195　　C.155.32.80.253　　D.155.32.80.191

【解析】155.32.80.192/26 表示 32 位长度的 IP 地址中，前 26 位是网络前缀，后 6 位是主机号，因此包含的主机地址个数为 $2^6-2=62$，主机地址范围为 155.32.80.193～155.32.80.254，显然 155.32.80.191 不属于这个网络。

【答案】(66)C；(67)D

【例6-7】校园网连接运营商的IP地址为202.117.113.3/30，本地网关的地址为192.168.1.254/24，如果本地计算机采用动态地址分配，在图6-5中应该如何配置？<u>(68)</u>。(2014年5月真题68)

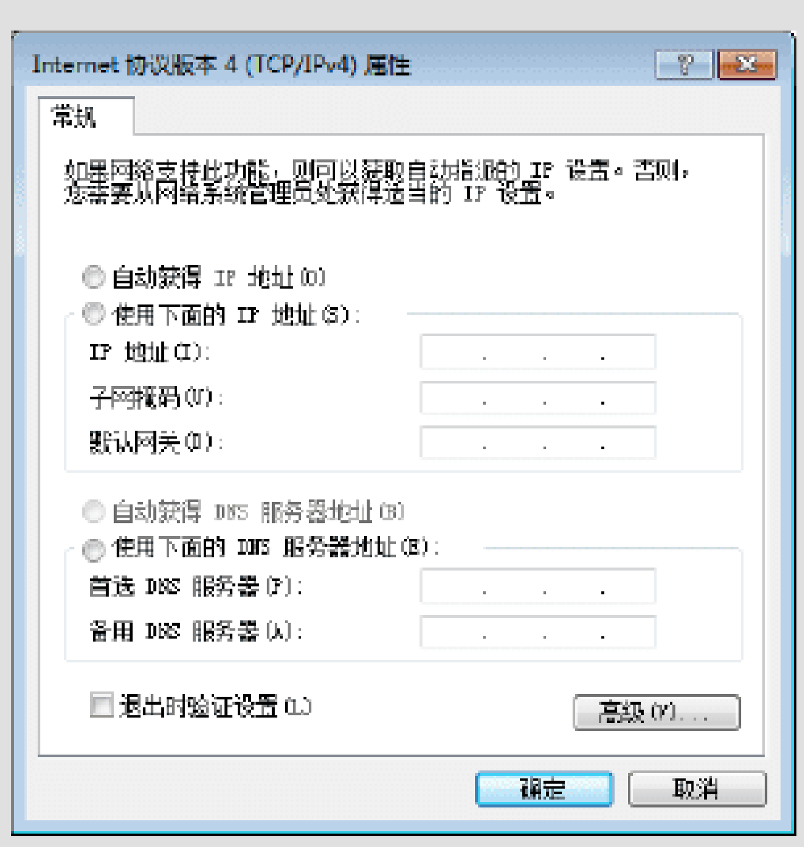


图 6-5　IP 设置

(68) A. 选取“自动获得 IP 地址”

B. 配置本地计算机 IP 地址为 192.168.1.×

C. 配置本地计算机 IP 地址为 202.115.113.×

D. 在网络 169.254.×.×中选取一个不冲突的 IP 地址

【解析】由于本地计算机采用动态地址分配，因此选用“自动获得 IP 地址”，DHCP 服务器会为本地计算机动态分配 IP 地址。

【答案】A

【例 6-8】在 IPv4 向 IPv6 的过渡期间，如果要使得两个 IPv6 结点可以通过现有的 IPv4 网络进行通信，则应该使用__(67)__；如果要使得纯 IPv6 结点可以与纯 IPv4 结点进行通信，则需要使用__(68)__。(2013 年 11 月真题 67～68)

(67) A. 堆栈技术　B. 双协议栈技术　C. 隧道技术　D. 翻译技术

(68) A. 堆栈技术　B. 双协议栈技术　C. 隧道技术　D. 翻译技术

【解析】因特网工程任务组(IETF)最早推荐从 IPv4 向 IPv6 过渡采用双栈技术和隧道技术，全世界很多运营商在不同规模上进行了 IPv6 的试验。隧道技术提供了一种以现有 IPv4 路由体系来传递 IPv6 数据的方法：将 IPv6 包作为无结构意义的数据，封装在 IPv4 包中，被 IPv4 网络传输。双协议栈技术是使 IPv6 节点与 IPv4 节点兼容的最直接方式，应用对象是主机、路由器等通信节点。支持双协议栈的 IPv6 节点与 IPv6 节点互通时使用 IPv6 协议栈，与 IPv4 节点互通时借助于 IPv4 over IPv6 技术使用 IPv4 协议栈。

实践表明，升级到双栈不仅没有给运营商带来直接的收益，反而影响了用户的体验。对于新建的 IPv6 网络，其用户数不可能与 IPv4 互联网上的用户数相比，如果 IPv6 的用户不能与 IPv4 的用户互联互通，则 IPv6 网络没有任何存在的价值。因此过渡的核心问题是新建 IPv6 网络必须与 IPv4 互联网互联互通。两种不同协议之间的互联互通，只能通过翻译技术解决，但是由于 IETF 在设计 IPv6 协议时，没有充分意识到与 IPv4 协议兼容的重要性，因此具有很高的技术难度。随着纯 IPv6 网络建设案例的增多和研究的深入，IETF 在 IPv4/IPv6 翻译技术，特别是无状态翻译技术方面取得了突破性进展，形成了系列 RFC 标准和工作组草案，为 IPv4 到 IPv6 过渡提供了新的技术方案。

【答案】(67)C；(68)D

6.4.2　Internet 服务

1. DNS 域名服务

Internet 中的域名地址和 IP 地址是等价的，它们之间是通过域名服务器来完成映射变换的。DNS 是一种分布式地址信息数据库系统，服务器中包含整个数据库的某部分信息，并供客户查询。域名系统采用的是客户机/服务器模式，整个系统由解析器和域名服务器组成。解析器是客户方，它负责查询域名服务器、解释从服务器返回来的应答、将信息返回给请求方等工作。域名服务器是服务器方，它通常保存着一部分域名空间的全部信息。

DNS 用的是 UDP 端口，端口号是 53。

2. 远程登录服务

远程登录服务是在 Telnet 协议的支持下，将用户计算机与远程主机连接起来，在远程

主机上运行程序，将相应的屏幕显示传送到本地机器，并将本地的输入送给远程计算机。

Telnet 协议用的是 TCP 端口，端口号一般为 23。

3. 电子邮件服务

电子邮件就是利用计算机进行信息交换的电子媒体信件。电子邮件地址的一般格式为：用户名@主机名。E-mail 系统基于客户机/服务器模式，整个系统由 E-mail 客户软件、E-mail 服务器和通信协议三部分组成。所用协议有简单邮件传送协议(SMTP)和用于接收邮件的 POP3 协议，两者均利用 TCP 端口，SMTP 所用的端口号是 25，POP3 所用的端口号是 110。

4. WWW 服务

WWW 服务是一种交互式图形界面的 Internet 服务，具有强大的信息连接功能。WWW 浏览程序为用户提供基于超文本传输协议(HTTP)的用户界面，WWW 服务器的数据文件由超文本标记语言(HTML)描述，HTTP 利用统一资源定位地址(URL)指向超媒体链接，并在文本内指向其他网络资源。一个 URL 包括：协议、主机域名、端口号(任选)、目录路径(任选)和一个文件名(任选)。其格式为 scheme://host.Domain[:port]Upath/filename。

WWW 使用的是一个众所周知的 TCP 端口，端口号为 80。

5. 文件传输服务

文件传输服务用来在计算机之间传输文件。FTP 是基于客户机/服务器模式的服务系统，它由客户软件、服务器软件和 FTP 通信协议三部分组成。FTP 在客户机与服务器的内部建立两条 TCP 连接：一条是控制连接，主要用于传输命令和参数(端口号为 21)；另一条是数据连接，主要用于传送文件(端口号为 20)。

真题链接

【例 6-9】 DHCP 客户端可以从 DHCP 服务器获得 (69) 。(2014 年 11 月真题 69)

(69) A. DHCP 服务器的地址和 Web 服务器的地址
B. DNS 服务器的地址和 DHCP 服务器的地址
C. 客户端地址和邮件服务器地址
D. 默认网关的地址和邮件服务器地址

【解析】 DHCP 客户端可以从 DHCP 服务器端获得 DHCP 服务器 IP 地址，DNS 服务器的 IP 地址，默认网关的 IP 地址。但是，不能获得 Web 服务器、邮件服务器的 IP 地址。

【答案】 B

【例 6-10】 POP3 协议采用 (69) 模式进行通信，当客户机需要服务时，客户端软件与 POP3 服务器建立 (70) 连接。(2013 年 11 月真题 69～70)

(69) A. Browser/Server　　B. Client/Server
C. Peer to Peer　　D. Peer to Server

(70) A. TCP　　B. UDP　　C. PHP　　D. IP

【解析】 POP 是适用于 C/S 结构的脱机模型的电子邮件协议，目前已发展到第 3 版，称为 POP3。POP3 是一种规定怎样将个人计算机连接到 Internet 的邮件服务器和下载电子邮件的电子协议，是因特网电子邮件的第一个离线标准。POP3 允许用户把服务器上的邮件存储到本地主机，同时删除先前保存在邮件服务

器上的邮件。POP3 服务器是遵循 POP3 协议的邮件接收服务器，是用于接收电子邮件的。电子邮件传输建立的是可靠连接，所以采用 TCP 连接。

【答案】(69)B；(70)A

6.5　网络安全

考核说明：与网络安全相关的题目主要出自第 1 章。本节主要介绍网络信息安全和防火墙技术，相对来说防火墙技术较为重要，但也不是考核要点，了解即可。

计算机网络安全是指计算机、网络系统的硬件、软件以及系统中的数据受到保护，不因偶然的或恶意的原因而遭到破坏、更改、泄露，确保系统能连续和可靠地运行，使网络服务不中断。广义地说，凡是涉及网络上信息的保密性、完整性、可用性、真实性和可控性的相关技术和理论，都是网络安全所要研究的领域。

网络安全涉及的主要内容包括运行系统的安全、信息系统的安全、信息传播的安全和信息内容的安全。信息系统对安全的基本需求有保密性、完整性、可用性、可控性和可核查性。

6.5.1　网络的信息安全

1. 信息的存储安全

信息的存储安全包括以下内容。

(1) 用户的标识与验证：限制访问系统的人员。

(2) 用户存取权限限制：限制进入系统的用户所能做的操作。

(3) 系统安全监控：建立一套安全监控系统，全面监控系统的活动。

(4) 病毒防治：网络服务器必须加装网络病毒自动检测系统。由于计算机病毒具有隐蔽性、传染性、潜伏性、触发性和破坏性等特点，因此需要建立计算机病毒防治管理制度。

(5) 数据的加密：防止非法窃取或调用。

(6) 计算机网络安全：通过采用安全防火墙系统、安全代理服务器、安全加密网关等实现网络信息安全的最外一层防线。

2. 信息的传输安全

信息的传输加密是面向线路的加密措施，有以下三种。

(1) 链路加密：只对两个结点之间的通信信道线路上所传输的信息进行加密保护。

(2) 结点加密：加、解密都在结点中进行，即每个结点中装有加、解密的保护装置，用于完成一个密钥向另一个密钥的转换。

(3) 端—端加密：为系统网络提供从信息源到目的地传送的数据的加密保护，可以是从主机到主机、终端到终端、终端到主机或到处理进程，或从数据的处理进程到处理进程，而不管数据在传送过程中经过了多少个中间结点，数据都不会被解密。

6.5.2　防火墙技术

所谓防火墙(Firewall)，是建立在内外网络边界上的过滤封锁机制，它认为内部网络是安全和可信赖的，外部网络是不安全和不可信赖的。防火墙的作用是防止不希望的、未经授权的数据包进出被保护的内部网络，通过边界控制强化内部网络的安全策略。

1. 防火墙的分类

可对防火墙进行如下分类。

(1) 包过滤型防火墙。工作在网络层，对数据包的源及目的 IP 具有识别和控制作用；对于传输层，它只能识别数据包是 TCP 还是 UDP 及所用的端口信息。

(2) 应用代理网关防火墙。彻底隔断内网与外网的直接通信，内网用户对外网的访问变成防火墙对外网的访问，然后再由防火墙转发给内网用户。

(3) 状态检测技术防火墙。结合了代理防火墙的安全性和包过滤防火墙的高速度等优点，在不损失安全性的基础上将代理防火墙的性能提高了 10 倍。

2. 典型防火墙的体系结构

一个防火墙系统通常是由过滤路由器和代理服务器组成。典型防火墙的体系结构包括包过滤路由器、双宿主主机、被屏蔽主机、被屏蔽子网等。

真题链接

【例 6-11】某用户在使用校园网中的一台计算机访问某网站时，发现使用域名不能访问该网站，但是使用该网站的 IP 地址可以访问该网站，造成该故障产生的原因有很多，其中不包括<u>(69)</u>。(2014 年 5 月真题 69)

(69) A. 该计算机设置的本地 DNS 服务器工作不正常　B. 该计算机的 DNS 服务器设置错误

C. 该计算机与 DNS 服务器不在同一子网　D. 本地 DNS 服务器网络连接中断

【解析】使用域名无法访问网站，而使用 IP 地址可以访问网站，说明域名无法解析成 IP 地址。大部分情况下，企业网的 DNS 服务器放置在防火墙的 DMZ 网段中，或者是企业网直接利用 Internet 中相关 DNS 服务器进行域名解析，它与作为客户端的计算机通常不在同一子网中。这一因素不会造成某台计算机不能使用域名访问外部服务器的现象。

【答案】C

6.6　多媒体概述

考核说明：多媒体和虚拟现实的基本概念，在最近几次考试中没有出现，不是考核重点，了解即可。相对来说，声音信号的数字化、声音文件格式较为重要，偶尔会考到。图形图像相关内容，在最近几次考试中经常出现，要掌握好，重点是图像数据量的计算、MPEG 国际标准和文件格式。动画和视频相关内容在考试中出现的概率不高，需要理解动画的本质、模拟视频的原理和数字视频的压缩方法。

6.6.1　多媒体的基本概念

媒体(Media)通常包括两方面的含义：一是指信息的物理载体(即存储和传递信息的实体)，如手册、磁盘、光盘、磁带以及相关的播放设备等；二是指承载信息的载体，即信息的表现形式(或者说传播形式)，如文字、声音、图像、动画和视频等，即 CCITT 定义的存储媒体和表示媒体。

1. 媒体的分类

可把媒体分为以下几类。

(1) 感觉媒体。指直接作用于人的感觉器官，使人产生直接感觉的媒体。

(2) 表示媒体。指传输感觉媒体的中介媒体，即用于数据交换的编码。

(3) 表现媒体。指进行信息输入和输出的媒体，如键盘、显示器等。

(4) 存储媒体。指用于存储表示媒体的物理介质，如硬盘、光盘等。

(5) 传输媒体。指传输表示媒体的物理介质，如电缆、光缆等。

2. 多媒体的概念和特性

多媒体就是多种信息载体的表现形式和连接方式。

多媒体技术的内涵极其广泛，所涉及的技术也很多，其主要特性有多样性、集成性、交互性、非线性、实时性、信息使用的方便性和信息结构的动态性。

真题链接

【例 6-12】以下媒体中__(12)__是表示媒体，__(13)__是表现媒体。(2014 年 11 月真题 12～13)

(12) A. 声音　　B. 声音编码　　C. 超声波　　D. 喇叭

(13) A. 声音　　B. 声音编码　　C. 超声波　　D. 喇叭

【解析】表示媒体是为了加工、处理和传输感觉媒体而人为研究、构造出来的一种媒体，它有各种编码方式，如文本编码、图像编码和声音编码等。

表现媒体是指进行信息输入和输出的媒体，如键盘、鼠标、扫描仪、话筒和摄像机等输入媒体以及显示器、打印机和扬声器等输出媒体。

感觉媒体是指直接作用于人的感觉器官，使人产生直接感觉的媒体，例如，引起听觉反应的声音，引起视觉反应的文本、图形和图像等。

存储媒体是指用于存储表示媒体的物理介质，如硬盘、软盘、光盘和胶卷等。

传输媒体是指用于存储表示媒体的物理介质，如电缆和光缆等。

通常在没有特殊说明的情况下，人们所说的信息媒体就是指信息的感觉媒体。

【答案】(12)B；(13)D

【例 6-13】以下媒体中__(12)__是表示媒体，__(13)__是表现媒体。(2014 年 5 月真题 12～13)

(12) A. 图像　　B. 图像编码　　C. 电磁波　　D. 鼠标

(13) A. 图像　　B. 图像编码　　C. 电磁波　　D. 鼠标

【解析】表示媒体指的是为了传输感觉媒体而人为研究出来的媒体，借助于此种媒体，能有效地存储

感觉媒体或将感觉媒体从一个地方传送到另一个地方，如语言编码、电报码、条形码等。

表现媒体指的是用于通信中使电信号和感觉媒体之间产生转换用的媒体。如输入、输出设备，包括键盘、鼠标器、显示器、打印机等。

【答案】(12)B；(13)D

6.6.2 虚拟现实的基本概念

虚拟现实是将现实世界的多维信息映射到计算机的数字空间生成相应的虚拟世界，主要包括基本模型构建、空间跟踪、声音定位、视觉跟踪和视点感应等关键技术。其主要特征有如下三个方面。

(1) 多感知。就是说除了一般计算机所具有的视觉感知外，还有听觉感知、力觉感知、触觉感知、运动感知，甚至包括味觉感知和嗅觉感知等。理想的虚拟现实就是应该具有人所具有的感知功能。

(2) 沉浸(又称临场感)。是指用户感到作为主角存在于模拟环境中的真实程度。理想的模拟环境应该达到使用户难以分辨真假的程度。

(3) 交互。是指用户对模拟环境内物体的可操作程度和从环境得到反馈的自然程度(包括实时性)。

6.6.3 声音

声音是通过空气传播的一种连续的波，即声波。声波在时间和幅度上都是连续的模拟信号，通常称为模拟声音(音频)信号。声音信号的两个基本参数是幅度和频率。

数字音频可分为波形声音和合成声音两种。

1. 声音信号的数字化

声音信号转换为数字信号的方法是，首先进行取样；对于取样的结果进一步地量化，得到数字形式的声音信号；为了便于计算机的存储、处理和传输，还必须按照一定的要求进行数据压缩和编码，即选择某一种或者几种方法对它进行压缩，以减少数据量，再按照某种规定的格式将数据组织成文件。

2. 波形声音

声音数字化后，其数据传输率与信号在计算机中的实时传输有直接关系，而其总数据量与计算机的存储空间有直接关系。未经压缩的数字音频数据传输率可按下式计算：

数据传输率(bps)=采样频率(Hz)×量化位数(b)×声道数

波形声音经过数字化后所需占用的存储空间可用如下公式计算：

声音信号数据量=数据传输率×持续时间/8(B)

3. 数字语音的压缩方法

数字语音的数据压缩方法很多，从原理上可分为三类：波形编码、参数编码和混合编码。

(1) 波形编码：一种直接对取样、量化后的波形进行压缩处理的方法。

(2) 参数编码：一种基于声音生成模型的压缩方法，从语音波形信号中提取生成的语音参数，使用这些参数通过语音生成模型重构出语音。

(3) 混合编码：上述两种方法的结合。

4. 声音合成

个人计算机和多媒体系统中的声音，除了数字波形声音之外，还有一类是使用符号表示的，由计算机合成的声音，包括语音合成和音乐合成。

- 语音合成目前主要指从文本到语音的合成，也称为文语转换。采用文语转换的方法输出语音，应预先建立语音参数数据库、发音规则库等。需要输出语音时，系统按需求先合成语音单元，再按语音学规律或者语音学规则连接成自然的语流。
- 音乐可以使用电子学原理合成出来(生成相应的波形)，各种乐器的音色也可以进行模拟。

5. MIDI

MIDI(Musical Instrument Digital Interface，乐器数字接口)，泛指数字音乐的国际标准。MIDI 标准规定了电子乐器与计算机连接的电缆硬件以及电子乐器之间、乐器与计算机之间传送数据的通信协议的规范；规定了音乐的数字表示，包括音符、定时、乐器指派等规范。

6. 声音文件格式

声音文件格式有如下几种。

(1) WAV 文件：Microsoft 公司的音频文件格式，它来源于对声音的模拟波形的采样。

(2) Module 文件：该格式的文件中存放乐谱和乐曲使用的各种音色样本，具有回放效果优异、音色种类无限等优点。

(3) MPEG 音频文件：现在最流行的声音文件格式，因其压缩率大，在网络可视电话通信方面应用广泛，但和 CD 唱片相比，音质不能令人非常满意。

(4) RealAudio 文件：具有强大的压缩量和较小的失真，主要目标是压缩比和容错性，其次才是音质。

(5) MIDI 文件：目前较成熟的音乐格式，实际上已经成为一种产业标准，其科学性、兼容性、复杂程度等各方面当然远远超过前面介绍的所有标准。

(6) Voice 文件：Creative 公司波形音频文件格式，也是声霸卡使用的音频文件格式。

(7) Sound 文件：Next Computer 公司推出的数字声音文件格式，支持压缩。

(8) Audio 文件：Sun Microsystems 公司推出的一种经过压缩的数字声音文件格式，是互联网上常用的声音文件格式。

(9) AIFF 文件：Apple 计算机的音频文件格式，Windows 的 Convert 工具可以把 AIF 格式的文件转换成 Microsoft 的 WAV 格式的文件。

(10) CMF 文件：Creative 公司的专用音乐格式，专用于 FM 声卡，兼容性较差。

真题链接

【例 6-14】声音信号数字化过程中首先要进行__(12)__。(2013 年 5 月真题 12)

(12) A. 解码　　B. D/A 转换　　C. 编码　　D. A/D 转换

【解析】声音信号是模拟信号，它不仅在幅度取值上是连续的，而且在时间上也是连续的。要使声音信号数字化并传递，首先要进行 A/D 转换。

【答案】D

6.6.4 图形和图像

1. 基本概念

1) 色彩三要素

为了确切地表示某一彩色光的度量，可以用亮度、色调和色饱和度三个物理量描述，称为色彩三要素。

- 亮度：描述光作用于人眼时引起的明暗程度感觉，是指彩色明暗深浅程度。
- 色调：指颜色的类别。
- 色饱和度：指某一颜色的深浅程度(或浓度)。

2) 三基色原理

从理论上讲，任何一种颜色都可以用三种基本颜色按不同比例混合得到。绝大多数颜色光可以分解成红、绿、蓝三种颜色光，这就是色度学中最基本的三基色原理。三基色的选择不是相互独立的，即任何一种颜色都不能由其他两种颜色合成。

3) 彩色空间

彩色空间是指彩色图像所使用的颜色描述方法，也称为彩色模型，主要有 RGB 彩色空间、CMY 彩色空间、YUV 彩色空间。

2. 计算机中的图形数据表示

1) 矢量图形

矢量图形是用一系列计算机指令来描述和记录的一幅图的内容，即通过指令描述构成一幅图的所有直线、曲线、圆、圆弧、矩形等图元的位置、位数和形状，也可以用更为复杂的形式表示图像中的曲面、光照、材质等效果。矢量图形实际上是用数学的方式来描述一幅图形图像，在处理图形图像时根据图元对应的数学表达式进行编辑和处理。

2) 位图图像

位图图像是指用像素点来描述的图像。位图图像在计算机内存中由一组二进制位组成，这些位定义图像中每个像素点的颜色和亮度。屏幕上的一个点也称为一个像素，显示一幅图像时，屏幕上的一个像素也就对应于图像中的某一个点。位图适合于表现比较细腻、层次较多、色彩较丰富、包含大量细节的图像，并可直接、快速地在屏幕上显示出来，但占用存储空间较大，一般需要进行数据压缩。

3. 图像的获取

图像的获取方式有如下几种。

- 利用数字图像库。
- 利用绘图软件创建图像。

- 利用数字转换设备采集图像。处理过程分为采样、量化和编码三步。

4. 图像的属性

1) 分辨率

- 显示分辨率：指显示屏上能够显示出的像素数目。
- 图像分辨率：指组成一幅图像的像素密度，也用水平和垂直的像素表示，即用每英寸多少点表示数字化图像的大小。

2) 图像深度

图像深度是指存储每个像素所用的位数，也用于量度图像的色彩分辨率。

3) 真彩色和伪彩色

真彩色是指组成一幅彩色图像的每个像素值中，有 R、G、B 三个基色分量，每个基色分量直接决定显示设备的基色强度，这样产生的彩色称为真彩色。在一些场合把 RGB 8∶8∶8 方式表示的彩色图像称为真彩色图像或全彩色图像。

图像中每个像素的颜色不是由 3 个基色分量的数值直接表达，而是把像素值作为地址索引，以便在彩色查找表中查找这个像素实际的 R、G、B 分量，人们将图像的这种颜色表达方式称为伪彩色。

5. 图形图像转换

1) 图形图像的硬件转换

一张工程图纸用扫描仪输入到 Photoshop，它就变成图像信息；用数字化仪将它输入到 AutoCAD 后，它就变成图形信息。当然可以先将图形用扫描仪扫入计算机，变成图像信息，再用一定的软件人工或自动地勾勒出它的轮廓，这个过程称为向量化，也就是图像转换为图形的过程。这个过程必然会丢失许多细节，所以通常用于工程绘图领域。

2) 图形图像的软件转换

较好的格式转换软件几乎提供所有常用图形图像文件格式之间的转换。较好的文件格式是指兼并图形图像各自优点的文件格式。

6. 图像的压缩编码

图像的分辨率越高，图像的深度越深，则数字化后的图像效果越逼真，图像数据量越大。估算数据量的公式为

$$图像数据量=图像的总像素\times图像深度/8(B)$$

其中，图像的总像素为图像的水平方向像素数乘以垂直方向像素数。

数据压缩可分成两类，一类是无损压缩，另一类是有损压缩。

- 无损压缩利用数据的统计冗余进行压缩，可以保证在数据压缩和还原过程中，图像信息没有损耗或失真，图像还原时可以完全恢复。
- 有损压缩方法利用人眼视觉对图像中的某些频率成分不敏感的特性，采用一些高效的有限失真数据压缩算法，允许压缩过程中损失一定的信息。

7. 多媒体数据压缩编码的国际标准

计算机中使用的图像压缩编码方法有多种国际标准和工业标准。目前使用相当广泛的

编码及压缩标准有 JPEG、MPEG 和 H.261。

(1) JPEG：是由 ISO 和 IEC 两个组织机构联合组成的专家组负责制定的静态数字图像数据压缩编码标准，这个专家组开发的算法称为 JPEG 算法，并且成为国际上通用的标准，因此又称为 JPEG 标准。

(2) MPEG：是由 ISO 和 IEC 两个组织机构联合组成的一个专家组制定的。第一阶段(MPEG-1)是针对普通电视质量的视频信号的压缩；第二阶段(MPEG-2)则是对高清晰度电视质量的信号进行压缩。目前又推出了许多多媒体内容描述接口标准等。每个新标准的产生都极大地推动了数字视频的发展和更广泛的应用。

(3) H.261：是国际电话电报咨询委员会提出的电话/会议电视的建议标准。

8. 图形、图像文件格式

(1) BMP 格式：BMP 是标准的 Windows 操作系统采用的图像文件格式，是一种与设备无关的位图格式，目的是让 Windows 能够在任何类型的显示设备上输出所存储的图像。

(2) GIF 格式：GIF 是压缩图像存储格式，它使用 LZW 压缩方法，压缩比较高，文件长度较小。支持黑白图像、16 色和 256 色的彩色图像。

(3) TIFF 格式：TIFF 格式是工业标准格式，支持所有图像类型。文件分成压缩和非压缩两大类。

(4) PCX 格式：PCX 格式是指使用 RLE 行程编码方法进行压缩的图像文件格式，支持黑白图像、16 色和 256 色的伪彩色图像、灰度图像以及 RGB 真彩色图像。

(5) JPEG 格式：JPEG 格式是指使用 JPEG 方法进行图像数据压缩。这种格式的最大特点是文件非常小。它是一种有损压缩的静态图像文件存储格式，支持灰度图像、RGB 真彩色图像和 CMYK 真彩色图像。

(6) PCD 格式：PCD 格式是 Photo-CD 的专用存储格式，文件中含有从专业摄影照片到普通显示用的多种分辨率的图像，所以数据量都非常大。

(7) 其他格式：包括 Targe、WMF、EPS 和 DIF 等。

真题链接

【例 6-15】显示深度、图像深度是图像显示的重要指标。当 (14) 时，显示器不能完全反映数字图像使用的全部颜色。(2014 年 11 月真题 14)

(14) A. 显示深度=图像深度　　B. 显示深度>图像深度
C. 显示深度>=图像深度　　D. 显示深度<图像深度

【解析】图像深度是图像文件中记录一个像素点所需要的位数，显示深度表示显示缓存中记录屏幕上一个点的位数，也即显示器可以显示的颜色数。因此，显示一幅图像时，屏幕上呈现的颜色效果与图文提供的颜色信息有关，也即与图像深度有关，同时也与显示器当前可容纳的颜色容量有关，也即与显示深度有关。

(1) 显示深度大于图像深度：在这种情况下屏幕上的颜色能较真实地反映图像文件的颜色效果。显示的颜色完全取决于图像的颜色。

(2) 显示深度等于图像深度：这种情况下如果用真彩色显示模式来显示真彩色图像，或者显示调色板与图像调色板一致时，屏幕上的颜色能较真实地反映图像文件的颜色效果；反之，显示调色板与图像调色

板不一致时，显示色彩会出现失真现象。

(3) 显示深度小于图像深度：此时显示的颜色会出现失真现象。

【答案】 D

【例 6-16】 (14) 表示显示器在横向(行)上具有的像素点数目。(2014 年 5 月真题 14)

(14) A. 显示分辨率　B. 水平分辨率　C. 垂直分辨率　D. 显示深度

【解析】 显示分辨率一般用显示设备水平方向和垂直方向上的最大像素数目来表示。水平分辨率表示显示器在横向(行)上具有的像素点数目；垂直分辨率则表示显示器在纵向(列)上具有的像素点数目。

【答案】 B

6.6.5　动画和视频

1. 基本概念

动画是将静态的图像、图形及图画等按一定时间顺序显示而形成连续的动态画面。计算机动画是采用连续播放静止图像的方法产生景物运动的效果，即使用计算机产生图形、图像运动的技术。动画的本质是运动。根据运动的控制方式可将计算机动画分为实时动画、矢量动画和三维动画三种。

1) 实时动画

实时动画是采用各种算法来实现运动物体的运动控制。采用的算法有运动学算法、动力学算法、反向运动学算法、反向动力学算法、随机动力学算法等。

2) 矢量动画

矢量动画是由矢量图衍生出的动画形式。矢量图是利用数学函数来记录和表示图形，矢量动画通过各种算法实现各种动画效果。

3) 三维动画

三维与二维动画的区别主要在于采用不同的方法获得动画中的景物运动效果。根据剧情的要求，首先要建立角色、实物和景物的三维数据模型；再对模型进行光照着色；然后使模型动起来，即模型可以在计算机控制下在三维空间中运动；最后对运动的模型重新生成图像再刷新屏幕，形成运动图像。

2. 模拟视频

1) 模拟视频的原理

电视信号通过光栅扫描的方法显示在荧光屏上，从荧光屏的顶部开始，一行一行地向下扫描，直至荧光屏的最底部，然后返回到顶部，重新开始扫描。这个过程产生的一个有序的图像信号的集合，组成了电视图像中的一幅图像，称为一帧；连续不断的图像序列就形成了动态视频图像。

2) 彩色电视的制式

世界上现行的彩色电视制式主要有 NTSC 制、PAL 制和 SECAM 制三种。采用隔行扫描比采用逐行扫描所占用的信号传输带宽要减少一半，这样有利于信道的利用，有利于信号传输和处理；采用每秒 25 帧的帧频能以最少的信号容量有效地满足人眼的视觉暂留特性；

采用 50Hz 的场频是因为我国的电网频率为 50Hz；采用 50Hz 的场刷新频率可有效地去掉电网信号的干扰。

3. 数字视频

对模拟视频信息进行数字化可采取如下方式。

(1) 先从复合彩色电视图像中分离出彩色分量，然后数字化。

(2) 先对全彩色电视信号数字化，然后在数字域中进行分离，以获得 YUV、YIQ 或 RGB 分量信号。

4. 数字视频标准

国际无线电咨询委员会(Consultative Committee of International Radio，CCIR)制定的广播级质量数字电视编码标准，即 CCIR601 标准，为 PAL、NTSC 和 SECAM 电视制式之间确定了公共的数字化参数。该标准规定了彩色电视图像转换成数字图像所使用的采样频率、采样结构、彩色空间转换等，这个标准对多媒体的开发和应用十分重要。

5. 视频压缩编码

视频压缩的目的是在尽可能保证视觉效果的前提下减少视频数据率。视频是连续的静态图像，其压缩编码算法与静态图像的压缩编码算法有某些共同之处，但是视频还有其自身的特性，在压缩时必须考虑其运动特性。

1) 无损压缩与有损压缩

无损压缩指压缩前和解压缩后的数据完全一致，多数的无损压缩都采用 RLE 行程编码算法。

有损算法意味着解压缩后的数据与压缩前不一致，在压缩的过程中要丢失一些人眼和人耳所不敏感的图像或声音信息，而且丢失的信息不可恢复。

2) 帧内和帧间压缩

帧内压缩也称为空间压缩。当压缩一帧视频时，仅考虑本帧的数据而不考虑相邻帧之间的冗余信息。

采用帧间压缩是由于许多视频或动画的连续前后两帧具有很大的相关性，或者说前后两帧信息变化很小。帧间压缩也叫时间压缩，它通过比较时间轴上不同帧之间的数据进行压缩。帧差值算法是一种典型的时间压缩法，它通过比较本帧与相邻帧之间的差异，仅记录本帧与其相邻帧的差值，这样可以大大减少数据量。

3) 对称和不对称编码

对称编码意味着压缩和解压缩需要相同的计算处理能力和时间。不对称或非对称意味着压缩时需要花费大量的处理能力和时间，而解压缩时则能较好地实时回放，即以不同的速度进行压缩和解压缩。

6. 视频文件格式

下面介绍视频文件格式。

(1) GIF 文件：采用无损压缩方法中效率较高的 LZW 算法，主要用于图像文件的网络传输。

(2) Flic 文件：采用行程编码算法和 Delta 算法进行无损的数据压缩，具有较高的数据压缩率。

(3) AVI 文件：AVI 文件允许视频和音频交错在一起播放，支持 256 色和 RLE 压缩，但 AVI 文件并未限定压缩标准。

(4) Quick Time 文件：该文件是 Apple 公司开发的一种音频、视频文件格式，用来保存音频和视频信息，具有先进的视频和音频功能。

(5) MPEG 文件：MPEG 文件是运动图像压缩算法的国际标准，它包括 MPEG 视频、MPEG 音频和 MPEG 系统三个部分。

(6) RealVideo 文件：RealVideo 文件是 Real Networks 公司开发的一种新型流式视频文件格式。

真题链接

【例 6-17】视频信息是连续的图像序列，__(11)__是构成视频信息的基本单元。(2013 年 11 月真题 11)

(11) A. 帧　　B. 场　　C. 幅　　D. 像素

【解析】视频信息是连续的图像序列，它是以帧为基本单元构成视频信息。

【答案】A

6.7　多媒体网络

考核说明： 本节主要介绍超文本与超媒体、流媒体的概念，在最近几次考试中没有出现，不是考核重点，了解即可。

6.7.1　超文本与超媒体

1. 超文本的概念

超文本是一种文本，与一般的文本文件的差别主要是组织方式不同，它将文本中遇到的一些相关内容通过链接组织在一起，用户可以很方便地阅览这些相关内容。超文本是一种文本管理技术，它以结点为单位组织信息，在结点与结点之间通过表示它们之间关系的链加以连接，构成特定内容的信息网络。结点、链和网络是超文本所包含的三个基本要素。

(1) 结点：超文本中存储信息的单元，由若干个文本信息块组成。

(2) 链：建立不同结点之间的联系。每个结点都有若干个指向其他结点或由其他结点指向自身的指针，该指针称为链。链通常是有向的，即从链源(源结点)指向链宿(目的结点)。链源可以是热字、热区、图元、热点或结点等，一般链宿都是结点。

(3) 网络：由结点和链组成的一个非单一、非顺序的非线性网状结构。

文本中的词、短语、图像、声音剪辑之间的链接，或者其他文件、超文本文件的链接，称为超链接(热链接)。

2. **超媒体的概念**

用超文本方式组织和处理的多媒体信息就是超媒体。超媒体不仅包含文字，而且还可以包含图形、图像、动画、声音和影视图像片断，这些媒体之间也是用超链接组织的，而且它们之间的链接也是错综复杂的。超媒体与超文本之间的不同是：超文本主要是以文字的形式表示信息，建立的链接关系主要是文字之间的链接关系；超媒体除使用文字外，还使用图形、图像、动画、声音或影视片断等多种媒体来表示信息，建立的链接是图形、图像、动画、声音或影视片断等多种媒体之间的链接关系。

6.7.2 流媒体

流媒体是指在网络中使用流式传输技术的连续时基媒体，而流媒体技术是指把连续的影像和声音信息经过压缩处理之后放到专用的流服务器上，让浏览者一边下载一边观看、收听，而不需要等到整个多媒体文件下载完成就可以即时观看和收听的技术。

一个流媒体系统一般由三部分组成：流媒体开发工具，用来生成流式格式的媒体文件；流媒体服务器组件，用来通过网络服务器发布流媒体文件；流媒体播放器，用于客户端对流媒体文件的解压和播放。目前应用比较广泛的流媒体系统主要有 Windows Media 系统、Real System 系统和 Quick Time 系统等。

6.8 多媒体计算机系统

考核说明：本节主要介绍多媒体计算机的硬件系统和软件系统，在最近几次考试中没有出现，了解即可。

通常将具有处理多种媒体的能力的计算机称为多媒体计算机(Multimedia Personal Computer，MPC)。

6.8.1 多媒体计算机硬件系统

多媒体硬件系统可以看成是在 PC 的基础上进行的硬件扩充，以适应多媒体信息处理功能的需求。计算机硬件及声像等媒体输入/输出设备构成了多媒体硬件平台。

多媒体计算机的主要硬件除了常规的硬件如主机、软盘驱动器、硬盘驱动器、显示器、打印机之外，还要有音频信息处理硬件、视频信息处理硬件及光盘驱动器等。

6.8.2 多媒体计算机软件系统

多媒体计算机软件系统主要包括多媒体操作系统、多媒体创作工具软件、多媒体素材编辑软件和多媒体应用软件。

1. 多媒体操作系统

多媒体操作系统必须具备如下功能：支持对多媒体环境下的各个任务进行管理和调度；支持多媒体应用软件运行；支持对多媒体声像及其他多媒体信息的控制和实时处理；支持多媒体的输入输出及相应的软件接口；支持对多媒体数据和多媒体设备的管理和控制以及图形用户界面管理等功能。也就是说它能够像一般操作系统处理文字、图形、文件那样去处理音频、图像、视频等多媒体信息，并能够对 CD-ROM 驱动器、录像机、MIDI 设备、数码相机、扫描仪等各种多媒体设备进行控制和管理。

2. 多媒体创作工具软件

多媒体创作工具软件是基于多媒体操作系统的系统软件，它提供了建立多媒体节目的构件和框架的功能，可以实现媒体的组接和交互跳转，如 PowerPoint、Director、Flash、Authorware 等。

3. 多媒体素材编辑软件

在多媒体应用中，很重要的一个环节是制作所需要的各种媒体素材。多媒体素材编辑软件用于采集、整理和编辑各种媒体数据，如 Photoshop、Illustrator、Freehand、3ds max、Premiere 等。

4. 多媒体应用软件

多媒体应用软件是具体实用的应用程序及演示软件，是直接面向用户或信息发送和接收的软件。这类软件直接与用户接口，用户只要根据应用软件所给出的操作命令，通过最简单的操作便可使用这些软件。

6.9　应试加油站

6.9.1　考频统计

通过对最近几次考试真题的分析，“网络和多媒体基础知识”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 6-1 所示。

表 6-1　历年考点统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：12～14、66～70	多媒体、显示分辨率、IP 地址、通信协议标准。	8
	下午：无		0
2014 年 5 月	上午：12～14、66～70	图像、多媒体、安全协议认证、网络协议、IP 地址、服务器	8
	下午：无		0
2013 年 11 月	上午：12～14、66～70	代理服务器、IPv4 向 IPv6 的过渡技术、POP3 协议、矢量图的概念、视频信息、多媒体素材编辑软件	8
	下午：无		0

续表

年 份	题 号	知 识 点	分值/分
2013 年 5 月	上午：12～14、66～70	路由器、交换机、子网划分、UDP 协议、DNS 服务器、声音数字化、分辨率、媒体的概念	8
	下午：无		0
2012 年 11 月	上午：12～14、66～70	Windows 服务器身份、ASP 对象、子网划分、故障分析、图像格式、流媒体、视频信号处理	8
	下午：无		0
2012 年 5 月	上午：12～14、66～70	交换机、SMTP 协议、网络的性能、综合布线、子网划分、图像分辨率、数字音乐合成技术	8
	下午：无		0

纵观历年试卷，本章知识点是以选择题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量大约为 3 道选择题，所占分值为 8 分(约占试卷总分值 75 分中的 10%)。本章试题主要考查 IP 地址、网络设备、TCP/IP 协议、多媒体基本概念，难度较低。IP 地址及子网划分、图像部分是考核的重点。

6.9.2 解题技巧

【典型题 6-1】网络配置如图 6-6 所示，其中使用了一台路由器、一台交换机和一台集线器，对于这种配置，下面的论断中正确的是＿(66)＿。(2013 年 5 月真题 66)

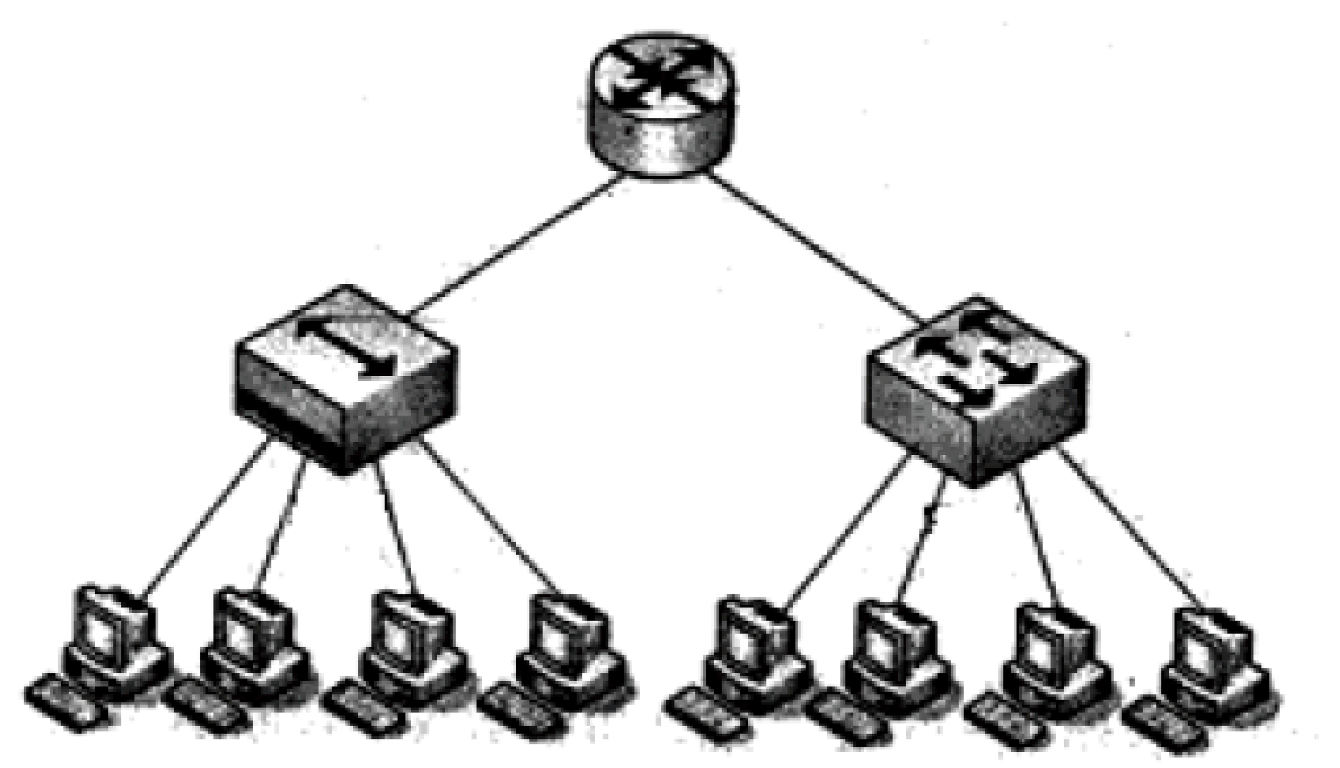


图 6-6 网络配置

(66) A. 2 个广播域和 2 个冲突域　　B. 1 个广播域和 2 个冲突域

C. 2 个广播域和 5 个冲突域　　D. 1 个广播域和 8 个冲突域

【解析】路由器工作在网络层，可以识别网络层的地址——IP 地址，有能力过滤第 3 层的广播消息。实际上，除非做特殊配置，否则路由器从不转发广播类型的数据包。因此，路由器的每个端口所连接的网络都独自构成一个广播域。

集线器上构成一个物理上的星型连接。但实际上，在集线器内部，各接口都是通过背板总线连接在一起的，在逻辑上仍构成一个共享的总线。因此，集线器和其所有接口所接的主机共同构成了一个冲突域和一个广播域。

交换机是通过为需要通信的两台主机直接建立专用的通信信道来增加可用带宽的。从这个角度上来讲，交换机相当于多端口网桥。交换机上的每个接口都是自己的一个冲突域。交换机和其所有接口所连接

的主机共同构成了一个广播域。

【答案】C

【典型题 6-2】以下关于网络中各种交换设备的叙述中，错误的是___(66)___。(2012 年 5 月真题 66)

(66) A. 以太网交换机根据 MAC 地址进行交换

B. 帧中继交换机只能根据虚电路号 DLCI 进行交换

C. 三层交换机只能根据第三层协议进行交换

D. ATM 交换机根据虚电路标识进行信元交换

【解析】三层交换机就是具有部分路由器功能的交换机，三层交换机的最重要目的是加快大型局域网内部的数据交换，所具有的路由功能也是为这个目的服务的，能够做到一次路由，多次转发。对于数据包转发等规律性的过程由硬件高速实现，而像路由信息更新、路由表维护、路由计算、路由确定等功能，由软件实现。三层交换技术就是二层交换技术+三层转发技术。传统交换技术是在 OSI 参考模型的第二层——数据链路层进行操作的，而三层交换技术是在网络模型中的第三层实现了数据包的高速转发，既可实现网络路由功能，又可根据不同网络状况做到最优网络性能。

【答案】C

【典型题 6-3】建筑物综合布线系统中的园区子系统是指___(69)___。(2012 年 5 月真题 69)

(69) A. 由终端到信息插座之间的连线系统　　B. 楼层接线间到工作区的线缆系统

C. 各楼层设备之间的互连系统　　D. 连接各个建筑物的通信系统

【解析】由终端到信息插座之间的连线系统是工作区子系统；楼层接线间到工作区的线缆系统是水平子系统；各楼层设备之间的互连系统是干线子系统。

【答案】D

【典型题 6-4】把网络地址 117.15.32.0/23 划分为 117.15.32.0/27，则得到的子网是_(67)_个。每个子网中可使用的主机地址是 _(68)_个。(2013 年 5 月真题 67～68)

(67) A. 4　　B. 8　　C. 16　　D. 32

(68) A. 30　　B. 31　　C. 32　　D. 34

【解析】题中将网络号为 23 位的 ip 地址划分为网络号为 27 位的子网，可以分为 $2^{(27-23)}=2^4=16$ 个子网。每个子网产生的主机地址是 $3^{(32-27)}-2=2^5-2=30$ 个。

【答案】(67)C；(68)A

【典型题 6-5】通常工作在 UDP 协议上的应用是_(69)_。(2013 年 5 月真题 69)

(69) A. 浏览网页　　B. Telnet 远程登录　　C. VoIP　　D. 发送邮件

【解析】浏览网页使用的是 HTTP 协议，而 HTTP 协议是搭载在 TCP 协议之上的。Telnet 协议是 TCP/IP 协议簇中的一员，是 Internet 远录登录服务的标准协议和主要方式。VoIP，即指在 IP 网络上使用 IP 协议以数据包的方式传输语音，使用 UDP 协议。

【答案】C

【典型题 6-6】随着网站知名度不断提高，网站访问量逐渐上升，网站负荷越来越重，针对此问题，一方面可通过升级网站服务器的软硬件来解决，另一方面可以通过集群技术，如 DNS 负载均衡来解决。在 Windows 的 DNS 服务器中通过 _(70)_ 操作可以确保域名解析并实现负载均衡。(2013 年 5 月真题 70)

(70) A. 启用循环，启动转发器指向每个 Web 服务器

B. 禁止循环，启动转发器指向每个 Web 服务器

C. 禁止循环，添加每个 Web 服务器的主机记录

D. 启用循环，添加每个 Web 服务器的主机记录

【解析】通过 DNS 服务器实现网络负载均衡是一种保证用户网络访问的方式。其方法有启用循环，添加每个 Web 服务器的主机记录。

【答案】D

【典型题 6-7】在 ASP 的内置对象中，__(67)__对象可以修改 cookie 中的值。(2012 年 11 月真题 67)

(67) A. request　　B. response　　C. application　　D. session

【解析】request 对象访问任何用 HTTP 请求传递的信息，包括 Form 表单用 POST 方法或 GET 方法传递的参数、cookie 和用户认证。response 对象控制发送给用户的信息，包括直接发送信息给浏览器、重定向浏览器到另一个 URL 或设置 cookie 的值。application 对象提供了存储空间用来存放变量和对象的作用。session 对象存储特定的用户会话所需的信息。

【答案】B

【典型题 6-8】分配给某公司网络的地址块是 220.17.192.0/20，该网络被划分为__(68)__个 C 类子网，不属于该公司网络的子网地址是__(69)__。(2012 年 11 月真题 68～69)

(68) A. 4　　B. 8　　C. 16　　D. 32

(69) A. 220.17.203.0　　B. 220.17.205.0　　C. 220.17.207.0　　D. 220.17.213.0

【解析】 这是一个 C 类地址块，其前 3 个字节表示网络地址，而“0/18”表示前 18 位为网络号，子网数占 24−20=4 位，子网个数为 2^4=16 个。子网地址最小为 220.17.193.0，最大为 220.17.207.0(207 的二进制对应为 11001111)，因此不属于该公司网络的子网地址是选项 D。

【答案】(68)C；(69)D

【典型题 6-9】SMTP 传输的邮件报文采用__(67)__格式表示。(2012 年 5 月真题 67)

(67) A. ASCII　　B. ZIP　　C. PNP　　D. HTML

【解析】SMTP(Simple Mail Transfer Protocol)是简单邮件传输协议，SMTP 协议仅限于传送 7 位的 ASCII 码，无法传送其他非英语国家的文字，也不能传输可执行文件和其他二进制文件。

【答案】A

【典型题 6-10】如果子网 172.6.32.0/20 被划分为子网 172.6.32.0/26，则下面的结论中正确的是__(70)__。(2012 年 5 月真题 70)

(70) A. 被划分为 62 个子网　　B. 每个子网有 64 个主机地址

C. 被划分为 31 个子网　　D. 每个子网有 62 个主机地址

【解析】172.6.32.0/26 说明划分后的子网地址中，主机位占 32-26=6 位，每个子网的主机地址个数为 2^6−2=62，之所以要减 2，是因为要去掉全 0 和全 1 的两个特殊地址。

【答案】D

【典型题 6-11】在 Windows 2003 Server 中启用配置 SNMP 服务时，必须以__(66)__身份登录才能完成 SNMP 服务的配置功能。(2012 年 11 月真题 66)

(66) A. guest　　B. 普通用户　　C. administrator 组成员　　D. user 组成员

【解析】在 Windows 2003 Server 中启用配置 SNMP 服务时，必须以 administrator 组成员身份登录才能完成 SNMP 服务的配置功能。

【答案】C

【典型题 6-12】如果 DNS 服务器更新了某域名的 IP 地址，造成客户端域名解析故障，在客户端可以用两种方法解决此问题，其中一种是在 Windows 命令行下执行__(70)__命令。(2012 年 11 月真题 70)

(70) A. ipconfig /all　　B. ipconfig /renew　　C. ipconfig /flushdns　　D. ipconfig /release

【解析】本题考查 Web 站点文档及相关知识。DNS 服务器更新了某域名的 IP 地址，造成客户端域名解析故障，需要使用刷新缓存的命令，即 ipconfig /flushdns。

【答案】C

【典型题 6-13】网络的可用性是指__(68)__。(2012 年 5 月真题 68)

(68) A. 网络通信能力的大小　　B. 用户用于网络维修的时间

C. 网络的可靠性　　D. 用户可利用网络时间的百分比

【解析】可用性不同于可靠性，可靠性是指在规定的条件下和指定的时间里，网络系统完成规定功能的能力。网络的可用性是指用户可利用网络时间的百分比。可用性达不到 100%，极高可用性在 99.999%以上，年度停机时间在 5 分钟以内要注意。

【答案】D

【典型题 6-14】计算机数字音乐合成技术主要有__(13)__两种方式，其中使用__(14)__合成的音乐，其音质更好。(2012 年 5 月真题 13～14)

(13) A. FM 和 AM　　B. AM 和 PM　　C. FM 和 PM　　D. FM 和 Wavetable

(14) A. FM　　B. AM　　C. PM　　D. Wavetable

【解析】数字音乐合成技术主要有FM Synthesis(调频合成法)和Wavetable Synthesis(波表合成法)。FM合成是通过振荡器产生正弦波，然后再叠加各种乐器的波形。由于振荡器成本较高，FM合成器提供的振荡器数目少，因此MIDI音乐听起来生硬呆板。波表合成则是采用真实的声音样本进行回放。声音样本汇总记录了各种真实乐器的波形采样，并保存在声卡的ROM或RAM中。通过波形合成的声音比FM合成的声音更为丰富和真实，但成本较高。

【答案】(13)D；(14)D

【典型题 6-15】以下关于 dpi 的叙述中，正确的是__(13)__。(2013 年 5 月真题 13)

(13) A. 每英寸的 bit 数　　B. 存储每个像素所用的位数

C. 每英寸像素点　　D. 显示屏上能够显示出的像素数目

【解析】dpi 是 Dots Per Inch 的缩写，是指每英寸的像素，也就是扫描精度。国际上都是计算一平方英寸面积内像素的多少。

【答案】C

【典型题 6-16】矢量图是常用的图形图像表示形式，__(10)__是描述矢量图的基本组成单位。(2013 年 11 月真题 10)

(10) A. 像素　　B. 像素点　　C. 图元　　D. 二进制位

【解析】矢量图在数学上定义为一系列由线连接的点，它的基本单元是图元。

【答案】C

【典型题 6-17】以下图像文件格式中，__(12)__可以在单个文件中实现多个图片轮流显示或者构成简单的动画效果。(2012 年 11 月真题 12)

(12) A. BMP　　B. JPEG　　C. GIF　　D. TIFF

【解析】BMP 是 Windows 操作系统采用的图像文件格式，是一种与设备无关的位图格式。

JPEG 是一种有损压缩的静态图像文件存储格式，压缩比很高。

GIF 以数据块为单位来存储图像的相关信息，它可以在一个文件中存放多幅彩色图像。如果逐幅读取存储于一个文件中的多幅图像，就可以像幻灯片那样显示或者构成简单的动画效果。

TIFF 是为扫描仪和桌面出版系统研制开发的一种较为通用的图像文件格式。

【答案】C

【典型题 6-18】使用___(12)___DPI 的分辨率扫描一幅 2×4 英寸的照片，可以得到一幅 300×600 像素的图像。(2012 年 5 月真题 12)

(12) A. 100　　B. 150　　C. 300　　D. 600

【解析】图像分辨率是组成一幅图像的像素密度，用每英寸多少点(dpi)表示数字化图像的大小。本题中，分辨率为300/2=150，或者600/4=150。

【答案】B

【典型题 6-19】计算机处理模拟视频信号过程中首先要进行_(14)_。(2012 年 11 月真题 14)

(14) A. A/D 变换　　B. 数据压缩　　C. D/A 变换　　D. 数据存储

【解析】计算机处理的是数字信号。因此，模拟信号进入计算机，首先要解决的是模拟视频信息的数字化问题，也就是将模拟信号转变为数字信号，这一过程为 A/D 变换。然后进行数据压缩和数据存储。

【答案】A

【典型题 6-20】媒体可以分为感觉媒体、表示媒体、表现媒体、存储媒体、传输媒体，_(14)_属于表现媒体。(2013 年 5 月真题 14)

(14) A. 打印机　　B. 硬盘　　C. 光缆　　D. 图像

【解析】表现媒体又称为显示媒体，是计算机用于输入输出信息的媒体，如键盘、鼠标、光笔、显示器、扫描仪、打印机、数字化仪等。

【答案】A

【典型题 6-21】以下多媒体素材编辑软件中，___(12)___主要用于动画编辑和处理。(2013 年 11 月真题 12)

(12) A. WPS　　B. Xara3D　　C. PhotoShop　　D. Cool Edit Pro

【解析】WPS 是文字编译系统；Xara3D 是 3D 文字动画制作工具；Photoshop 主要处理以像素所构成的数字图像；Cool Edit Pro 是一款音频编辑器。

【答案】B

【典型题 6-22】计算机终端用户因等待时间过程而放弃下载或显示图片/影像的问题，不能通过采用_(13)_来解决。(2012 年 11 月真题 13)

(13) A. 流媒体技术　　B. JPEG 累进编码方式

C. GIF 图像文件格式　　D. AD/DA 编码方式

【解析】AD/DA 是模数/数模转换，用来进行数字信号和模拟信号之间的转换，不能用来解决网络中图像传输和显示的问题。

流媒体技术就是把连续的影像和声音信息经过压缩处理后放在网站服务器上，让用户一边下载一边观看、收听，而不需要等整个压缩文件都下载到自己的计算机上才可以观看的网络传输技术。

采用 JPEG 累进编码时，图像编码在多次扫描中完成。累进编码传输时间长，接收端收到的图像是多次扫描由粗糙到清晰的累进过程。

GIF 格式采用无损压缩技术，只要图像不多于 256 色，就既可减少文件的大小，又可保持成像的质量。由于 GIF 格式支持透明背景图像，适用于多种操作系统，而且“体型”很小，所以网上很多小动画都是 GIF 格式。

【答案】D

6.10　过关习题

1. “三网合一”的三网是指______。

   A. 电信网、广播电视网、互联网　　B. 物联网、广播电视网、电信网

   C. 物联网、广播电视网、互联网　　D. 物联网、电信网、互联网

2. 下面关于集线器与交换机的描述中，错误的是______。

   A. 交换机是一种多端口网桥

   B. 交换机的各个端口形成一个广播域

   C. 集线器的所有端口组成一个冲突域

   D. 集线器可以起到自动寻址的作用

3. 帧中继作为一种远程接入方式有许多优点，下面的选项中错误的是______。

   A. 帧中继比 X.25 的通信开销少，传输速度更快

   B. 帧中继与 DDN 相比，能以更灵活的方式支持突发式通信

   C. 帧中继比异步传输模式能提供更高的数据速率

   D. 租用帧中继虚电路比租用 DDN 专线的费用低

4. ARP 属于__(1)__协议，它的作用是__(2)__。

(1) A. 物理层　　B. 数据链路层

   C. 网络层　　D. 传输层

(2) A. 实现 MAC 地址与主机名之间的映射

   B. 实现 IP 地址与 MAC 地址之间的变换

   C. 实现 IP 地址与端口号之间的映射

   D. 实现应用进程与物理地址之间的变换

5. TCP 是互联网中的__(1)__协议，使用__(2)__次握手协议建立连接。

(1) A. 传输层　　B. 网络层　　C. 会话层　　D. 应用层

(2) A. 1　　B. 2　　C. 3　　D. 4

6. A 类网络是很大的网络，每个 A 类网络中可以有__(1)__个网络地址。实际使用中必须把 A 类网络划分为子网，如果指定的子网掩码为 255.255.192.0，则该网络被划分为__(2)__个子网。

(1) A. 210　　B. 212　　C. 220　　D. 224

(2) A. 128　　B. 256　　C. 1024　　D. 2048

7. 要使 4 个连续的 C 类网络汇聚成一个超网，则子网掩码应该为______。

   A. 255.240.0.0　　B. 255.255.0.0

   C. 255.255.252.0　　D. 255.255.255.252

8. 在 Windows 系统中，为排除 DNS 域名解析故障，需要刷新 DNS 解析器缓存，应使用的命令

是______。

A. ipconfig/renew　　B. ipconfig/flushdns

C. netstat -r　　D. arp -a

9. HTTP 中，用于读取一个网页的操作方法为______。

A. READ　　B. GET

C. HEAD　　D. POST

10. 计算机通过 MIC(话筒接口)收到的信号是______。

A. 音频数字信号　　B. 音频模拟信号

C. 采样信号　　D. 量化信号

11. 一幅彩色图像(RGB)，分辨率为 256×512，每一种颜色用 8b 表示，则该彩色图像的数据量为______b。

A. 256×512×8　　B. 256×512×3×8　　C. 256×512×3/8　　D. 256×512×3

12. 以下编码方法中，______属于熵编码。(2011 年 5 月上午试题 12)

A. 哈夫曼编码　　B. 小波变换编码　　C. 线性预测编码　　D. 行程编码

13. ______既不是图像编码也不是视频编码的国际标准。

A. JPEG　　B. MPEG　　C. ADPCM　　D. H.261

14. 由 ISO 制定的 MPEG 系列标准中，______是多媒体内容描述接口标准。

A. MPEG-1　　B. MPEG-2　　C. MPEG-4　　D. MPEG-7

15. CIF 视频格式的图像分辨率为______。

A. 352×240　　B. 352×288　　C. 640×480　　D. 320×240

16. 多媒体制作过程中，不同媒体类型的数据收集、制作需要不同的软、硬件设备和技术手段，动画制作一般通过______进行。

A. 字处理软件　　B. 视频卡　　C. 声卡　　D. 图形/图像软件

第 7 章

数据库技术

本章要点

- 数据库管理系统(DBMS)
- 数据库的三级模式结构
- 数据模型
- E-R 模型
- 关系代数
- 关系数据库 SQL
- 函数依赖
- 规范化
- 模式分解
- 设计关系模式
- 数据库需求分析
- 数据库设计

7.1 基 本 概 念

考核说明：本节主要介绍数据库和数据库管理系统的基本概念、数据库系统的体系结构、数据库的三级模式结构等内容，在最近几次考试中没有涉及相关的考题，但 DBMS 的功能和数据库的三级模式较为重要，要能理解。

7.1.1 数据库系统

数据库系统(Database System，DBS)从广义上讲是由数据库、硬件、软件和人员组成的，其中管理的对象是数据。数据是经过组织的比特的集合，而信息是具有特定释义和意义的数据。

7.1.2 数据库管理系统

数据库管理系统(Database Management System，DBMS)是数据库系统的核心软件，要在操作系统的支持下工作，解决如何科学地组织和存储数据、如何高效地获取和维护数据库的系统软件问题。其主要功能包括数据定义功能、数据操纵功能、数据库的运行管理和数据库的建立与维护。

7.1.3 数据库的三级模式结构

数据库系统采用三级模式结构，这是数据库管理系统内部的系统结构。

- 概念模式(Schema)：也称模式，是数据库中全体数据的逻辑结构和特征的描述，它由若干个概念记录类型组成，只涉及行的描述，不涉及具体的值。概念模式的一个具体值称为模式的一个实例，同一个模式可以有很多实例。
- 外模式(External Schema)：也称用户模式或子模式，是用户与数据库系统的接口，是用户用到的那部分数据的描述，由若干个外部记录类型组成。描述外模式的数据定义语言称为外模式 DDL。
- 内模式(Internal Schema)：也称存储模式，是数据物理结构和存储方式的描述，是数据在数据库内部的表示方式，定义所有的内部记录类型、索引和文件的组织方式，以及数据控制方面的细节。描述内模式的数据定义语言称为内模式 DDL。

7.1.4 两级映像

数据库系统在三级模式之间提供了两级映像：模式/内模式映像、外模式/模式映像。

- 模式/内模式映像：该映像存在于概念级和内部级之间，实现了概念模式到内模式

之间的相互转换。

- 外模式/模式映像：该映像存在于外部级和概念级之间，实现了外模式到概念模式之间的相互转换。

DBMS 的二级映像功能保证了数据的独立性。

7.2　数 据 模 型

考核说明：本节主要介绍数据模型的基本概念、E-R 模型、层次模型、网状模型和关系模型等内容，其中 E-R 模型较为重要，需要理解和掌握。

7.2.1　数据模型的基本概念

模型就是对现实世界特征的模拟和抽象，数学模型是对现实世界数据特征的抽象。从事物的客观特性到计算机中的具体表示经历了现实世界、信息世界和机器世界三个数据领域。

(1) 概念数据模型。也称信息模型，是按用户的观点对数据和信息建模，是现实世界到信息世界的第一层抽象，强调其语义表达功能，易于用户理解，是用户和数据库设计人员交流的语言，主要用于数据库设计。这类模型中最著名的是实体联系模型，简称 E-R 模型。

(2) 基本数据模型。它是按计算机系统的观点对数据建模，是现实世界数据特征的抽象，用于 DBMS 的实现。基本的数据模型有层次模型、网状模型、关系模型和面向对象模型(Object-oriented Model)。

数据模型是用来描述数据的一组概念和定义。数据模型的三要素是数据结构、数据操作、数据的约束条件。

1. 层次模型

层次模型采用树型结构表示数据与数据之间的联系。在层次模型中，每一个结点表示一个记录类型(实体)，记录之间的联系用结点之间的连线表示，根结点以外的其他结点有且仅有一个双亲结点。

2. 网状模型

采用网状结构表示数据与数据间的联系的数据模型称为网状模型。在网状模型中，允许一个以上的结点无双亲，一个结点可以有多于一个的双亲。

网状模型中的每个结点表示一个记录类型(实体)，每个记录类型可以包含若干个字段(实体的属性)，结点间的连线表示记录类型之间一对多的联系。

3. 关系模型

关系数据库系统采用关系模型作为数据的组织方式，在关系模型中用表格结构表达实体集以及实体集之间的联系，其最大特色是描述的一致性。关系模型是由若干个关系模式组成的集合。一个关系模式相当于一个记录型，对应于程序设计语言中类型定义的概念。

7.2.2 E-R 模型

实体-联系模型简称 E-R 模型，所采用的三个主要概念是实体、联系和属性。E-R 模型是软件工程设计中的一个重要方法，因为它接近于人的思维方式，容易理解并且与计算机无关，所以用户容易接受。一般遇到实际问题，应先设计一个 E-R 模型，然后再把它转换成计算机能接受的数据模型。

1. 实体

实体是现实世界中可以区别于其他对象的“事件”或“物体”。每个实体由一组特性(属性)来表示，其中的某一部分属性可以唯一表示实体。实体集是具有相同属性的实体集合。

2. 联系

实体集之间的对应关系称为联系。实体的联系分为实体内部的联系和实体与实体之间的联系。实体集内部的联系反映数据在同一记录内部各字段间的联系。实体集之间的联系类型有一对一联系、一对多联系和多对多联系。

多个实体集间的联系类型有多个实体集间的一对一联系、多个实体集间的一对多联系和多个实体集间的多对多联系。

同一个实体集内部的各实体之间也存在 1∶1、1∶n 和 m∶n 的联系。

3. 属性

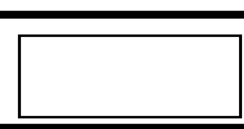
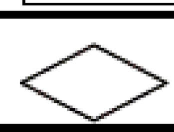
属性是实体某方面的特性。在同一实体集中，每个实体的属性及其域是相同的，但可能取不同的值。E-R 模型中的属性有如下分类。

- 简单属性和复合属性：简单属性是原子的、不可再分的，复合属性可以细分为更小的部分(即划分为别的属性)。
- 单值属性和多值属性：若定义的属性对于一个特定的实体只有单独的一个值，这样的属性叫作单值属性；若定义的属性对应一组值，则称为多值属性。
- NULL 属性：当实体在某个属性上没有值或属性值未知时，使用 NULL 值，表示无意义或不知道。
- 派生属性：可以从其他属性得来。

4. E-R 方法

概念模型中最常用的方法是实体-模型方法，简称 E-R 方法。该方法直接从现实世界中抽象出实体和实体间的联系，然后用非常直观的 E-R 图来表示数据模型。在 E-R 图中有如表 7-1 所示的几个主要构件。

表 7-1　E-R 图中的主要构件

构 件 名	构件符号	说　明
矩形		表示实体集
菱形		表示联系集

续表

构 件 名	构件符号	说　明
椭圆		表示属性
线段		将属性与相关的实体集连接，或将实体集与联系集相连
双椭圆		表示多值属性
虚椭圆		表示派生属性
双线		表示一个实体全部参与到联系集中

5. 扩充的 E-R 模型

扩充的 E-R 模型包括弱实体、特殊化、概括、聚集等概念。

7.3　关 系 代 数

考核说明：本节主要介绍关系数据库的基本概念、关系代数运算及扩展的运算等内容，最近几次考试中的相关考题不多，要重点掌握关系模式和主码的确定、基本的关系代数运算。

7.3.1　关系数据库的基本概念

1. 属性和域

在现实世界中，要描述一个事物，常常取其若干特征来表示。这些特征称为属性。每个属性的取值范围对应一个值的集合，称为该属性的域。

一般在关系数据库模型中，对域还加了一个限制，即所有的域都应是原子数据的集合。关系数据模型的这种限制称为第一范式(1NF)条件。如果关系数据模型突破了 1NF 的限制，则称为非 1NF 的。

2. 笛卡儿积与关系

【定义 7-1】 设 $D_1,D_2,\cdots,D_n$ 为任意集合，定义 $D_1,D_2,\cdots,D_n$ 的笛卡儿积为

$$D_1 \times D_2 \times \cdots \times D_n = \{(d_1,d_2,\cdots,d_n) \mid d_i \in D_i, i=1,2,\cdots,n\}$$

其中，每一个元素$(d_1,d_2,\cdots,d_n)$叫作一个 n 元组，元组的每一个值 d_i 叫作元组的一个分量，若 $D_i(i=1, 2, \cdots, n)$为有限集，其基数为 m_i，$i=1, 2, \cdots, n$，则 $D_1\times D_2\times\cdots\times D_n$ 的基数 M 为

$$M = \prod_{i=1}^{n} m_i$$

笛卡儿可以用二维表来表示。

【定义 7-2】 $D_1\times D_2\times\cdots\times D_n$ 的子集叫作在域 $D_1, D_2, \cdots, D_n$ 上的关系，记为

$$R(D_1,D_2,\cdots,D_n)$$

称关系 R 为 n 元关系。

从定义 7-2 可以得出一个关系也可以用二维表来表示。关系中属性的个数称为元数，元组的个数称为基数。

3. 关系的相关名词

下面介绍关系的相关名词。

(1) 目或度：n 元关系 $R(D_1,D_2,\cdots D_n)$，R 表示关系的名字，n 表示关系的目或度，也就是一张关系表中属性列的数目。

(2) 候选码：若关系中的某一属性或属性组的值能唯一标识一个元组，则称该属性或属性组为候选码。

(3) 主码：若一个关系有多个候选码，则选定其中一个为主码。

(4) 主属性：包含在任何候选码中的诸属性称为主属性。不包含在任何候选码中的属性称为非码属性。

(5) 外码：如果关系模式 R 中的属性或属性组非该关系的码，但却是其他关系的码，那么该属性集对关系模式 R 而言是外码。

(6) 全码：关系模式的所有属性组是这个关系模式的候选码，称为全码。

4. 关系的三种类型

下面介绍关系的三种类型。

(1) 基本关系(通常又称为基本表、基表)：是实际存在的表，是实际存储数据的逻辑表示。

(2) 查询表：查询结果对应的表。

(3) 视图表：由基本表或其他视图表导出的表。由于它本身不独立存储在数据库中，数据库中只存放它的定义，因此常称为虚表。

5. 关系数据库模式

在数据库中要区分型和值。关系数据库中的型也称为关系数据库模式，是关系数据库结构的描述，它包括若干域的定义以及在这些域上定义的若干关系模式。关系数据库的值是这些关系模式在某一时刻对应的关系的集合，通常称为关系数据库。

【定义 7-3】 关系的描述称为关系模式。可以形式化地表示为

$$R(U, D, \mathrm{dom}, F)$$

式中：R 为关系名；U 为组成该关系的属性名集合；D 为属性的域；dom 为属性向域的映像集合；F 为属性间数据的依赖关系集合。

通常将关系模式简记为

$$R(U) \text{或} R(A_1, A_2, \cdots, A_n)$$

式中：R 为关系名；$A_1, A_2, \cdots, A_n$ 为属性名或域名，属性向域的映像常常直接说明属性的类型、长度。通常为关系模式主属性加下划线来表示该属性为主码属性。

6. 完整性约束

完整性规则提供了一种手段来保证当授权用户对数据库作修改时不会破坏数据的一致性，因此，完整性规则防止的是对数据的意外破坏。关系模型的完整性规则是对关系的某

种约束条件。完整性共分为三类：实体完整性、参照完整性(也称引用完整性)和用户定义完整性。

(1) 实体完整性：规定基本关系 R 的主属性 A 不能取空值。

(2) 参照完整性：现实世界中的实体之间往往存在某种联系，在关系模型中实体与实体之间的联系是用关系来描述的，这样自然就存在着关系与关系间的引用。参照完整性规定，若 F 是基本关系 R 的外码，它与基本关系 S 的主码相对应(基本关系 R 和 S 不一定是不同的关系)，则对于 R 中每个元组在 F 上的值必须为：或者取空值(F 的每个属性值均为空值)，或者等于 S 中某个元组的主码值。

(3) 用户定义完整性：就是针对某一具体的关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求，由应用的环境决定。

7.3.2 五种基本的关系代数运算

五种基本的关系代数运算是并、差、笛卡儿积、投影和选择，其他运算可以通过基本的关系运算导出。

1. 并

关系 R 与 S 具有相同的关系模式，即 R 与 S 的元数相同(结构相同)。关系 R 和关系 S 的并(Union)由属于 R 或属于 S 的元组构成的集合组成，记作

$$R \cup S = \{t \mid t \in R \vee t \in S\}$$

式中：t 为元组变量。

2. 差

关系 R 与 S 具有相同的关系模式。关系 R 与 S 的差(Difference)由属于 R 但不属于 S 的元组构成的集合组成，记作

$$R - S = \{t \mid t \in R \wedge t \notin S\}$$

3. 广义笛卡儿积

两个元数分别为 n 目和 m 目的关系 R 和 S 的广义笛卡儿积(Extended Cartesian Product)是一个(n+m)列的元组的集合。元组的前 n 列是关系 R 的一个元组，后 m 列是关系 S 的一个元组。若 R 有 k_1 个元组，S 有 k_2 个元组，则关系 R 和 S 的广义笛卡儿积有 $k_1 \times k_2$ 个元组。记作

$$R \times S = \{t \mid t = < t_n, t_m > \wedge t_n \in R \wedge t_m \in S\}$$

4. 投影

投影(Projection)运算是从关系的垂直方向进行运算，在关系 R 中选择出若干属性列 A 组成新的关系。记作

$$\pi_A(R) = \{t[A] \mid t \in R\}$$

5. 选择

选择(Selection)运算是从关系的水平方向进行运算，是从关系 R 中选择满足给定条件的诸元素，记作

$$\sigma_F(R)=\{t\mid t\in R\wedge F(t)=\text{true}\}$$

其中，F 中的运算对象是属性名(或列的序号)或常数，运算符是算术比较符和逻辑运算符。

7.3.3 扩展的关系代数运算

扩展的关系代数运算可以从基本的关系运算中导出，主要包括以下几种。

1. 交

关系 R 与 S 具有相同的关系模式。关系 R 和 S 的交由属于 R 同时又属于 S 的元组构成的集合组成。关系 R 和 S 的交记作 $R\cap S$。

2. 连接

连接(Join，也作联接)分为 θ 连接、等值连接和自然连接三种。连接运算是从两个关系的笛卡儿积中选取满足条件的元组。

(1) θ 连接：从关系 R 和 S 的笛卡儿积中选取属性间满足一定条件的元组。记作

$$R\underset{X\theta Y}{\bowtie}S$$

(2) 等值连接：当 θ 为“=”时称为等值连接。

(3) 自然连接：是一种比较特殊的等值连接，它要求两个关系中进行比较的分量必须是相同的属性组，并且在结果集中把重复属性列去掉。

3. 除

除(Division)运算是同时从关系的水平方向和垂直方向进行运算。给定关系 $R(X, Y)$和 $S(Y, Z)$，X、Y、Z 为属性组。$R\div S$ 应当满足元组在 X 上的分量值 x 的象集 Y_x 包含关系 S 在属性组 Y 上投影的元素的集合。

4. 广义投影

广义投影(Generalized Projection)运算允许在投影列表中使用算术运算，实现了对投影运算的扩充。

若有关系 R，条件 F_1, F_2, …, F_n 中的每一个都是涉及 R 中常量和属性的算术表达式，那么广义投影运算的形式定义如下：

$$\pi_{F_1,F_2,\cdots,F_n}(R)$$

5. 外连接

外连接(Outer Join)运算是连接运算的扩展，可以处理缺失的信息。外连接运算有三种：左外连接、右外连接和全外连接。

左外连接：取出左侧关系中所有与右侧关系中任一元组都不匹配的元组，用空值 NULL

来填充所有来自右侧关系的属性，构成新的元组，将其加入自然连接的结果中。

右外连接：取出右侧关系中所有与左侧关系中任一元组都不匹配的元组，用空值 NULL 来填充所有来自左侧关系的属性，构成新的元组，将其加入自然连接的结果中。

全外连接：完成左外连接和右外连接的操作。即填充左侧关系中所有与右侧关系中任一元组都不匹配的元组，填充右侧关系中所有与左侧关系中任一元组都不匹配的元组，将产生的新元组加入自然连接的结果中。

真题链接

【例 7-1】关系 R1 和 R2 如图 7-1 所示。

R1

A	B	C	D
a	d	c	e
c	b	a	e
d	e	c	e
e	f	d	a

R2

C	D	E	F
a	e	c	a
a	e	a	b
c	e	b	c

图 7-1　关系 R1 和 R2

若进行 R1⋈R2 运算，则结果集为 (51) 元关系，共有 (52) 个元组。(2012 年 11 月真题 51～52)

(51) A. 4　　B. 5　　C. 6　　D. 7

(52) A. 4　　B. 5　　C. 6　　D. 7

【解析】R1⋈R2 即 R1 与 R2 作自然连接运算，要求两个关系中进行比较的分量必须是相同的属性组，并且在结果集中将重复属性列去掉。进行 R1⋈R2 运算后，结果集的属性列应为 ABCDEF，即 6 元关系，运算结果如表 7-2 所示，共有 4 个元组。

表 7-2　运算结果

A	B	C	D	E	F
a	d	c	e	b	c
c	b	a	e	c	a
c	b	a	e	a	B
d	e	c	e	b	c

【答案】(51)C；(52)A

7.4　关系数据库 SQL 简介

考核说明：本节主要介绍 SQL 数据库体系结构和各种 SQL 语句，考试中会考到常用的 SQL 语句，要记住各种语句的格式。

7.4.1 SQL 数据库体系结构

SQL(Structured Query Language)是在关系数据库中应用最普遍的语言，它不仅包含数据查询功能，还包括插入、删除、更新和数据定义功能。

SQL 的基本组成如下。

(1) 数据定义语言(DDL)：SQL DDL 提供定义关系模式和视图、删除关系和视图、修改关系模式的命令。

(2) 交互式数据操纵语言(DML)：SQL DML 提供查询、插入、删除和修改的命令。

(3) 事务控制(Transaction Control)：SQL 提供的定义事务开始和结束的命令。

(4) 嵌入式 SQL 和动态 SQL：用于嵌入到某种通用的高级语言(C、C++、Java、PL/I、Cobol、VB 等)中混合编程。其中 SQL 负责操纵数据库，高级语言负责控制程序流程。

(5) 完整性(Integrity)：SQL DDL 包括定义数据库中的数据必须满足的完整性约束条件的命令，对于破坏完整性约束条件的更新将被禁止。

(6) 权限管理(Authorization)：SQL DDL 中包括说明对关系和视图的访问权限的命令。

7.4.2 SQL 数据定义

1. 创建表

语句格式：

```
CREATE TABLE <表名>(<列名><数据类型>[列级完整性约束条件]
                        [,<列名><数据类型>[列级完整性约束条件]]…
                        [,<表级完整性约束条件>]);
```

列级完整性约束条件有 NULL、UNIQUE，如 NOT NULL UNIQUE 表示取值唯一，不能取空值。

2. 修改表和删除表

1) 修改表

语句格式：

```
ALERT TABLE <表名>[ADD<新列名><数据类型>[列级完整性约束条件]]
       [DROP <完整性约束名>]
       [MODIFY <列名><数据类型>];
```

2) 删除表

语句格式：

```
DROP TABLE <表名>
```

3. 定义和删除索引

1) 建立索引

语句格式：

```
CREATE [UNIQUE] [CLUSTER] INDEX <索引名>
    ON <表名>(<列名>[<次序>][,<列名>[<次序>]]…);
```

参数说明如下。

- <次序>：可选升序(ASC)或降序(DSC)。默认值为 ASC。
- UNIQUE：表明此索引的每一个索引值只对应唯一的数据记录。
- CLUSTER：表示要建立的索引是聚集索引，意为索引项的顺序与表中记录的物理顺序一致。

2)　删除索引

语句格式：

```
DROP INDEX <索引名>;
```

4. 视图创建与删除

视图是从一个或多个表或视图中导出的表，其结构和数据是建立在对表查询的基础上的。视图不是真实存在的基础表而是一个虚拟表，视图所对应的数据并不实际地以视图结构存储在数据库中，而是存储在视图所引用的表中。

1)　视图的创建

语句格式：

```
CREATE VIEW 视图名 (列表名)
    AS SELECT 查询子句
    [WITH CHECK OPTION];
```

2)　视图的删除

语句格式：

```
DROP VIEW 视图名;
```

7.4.3　SQL 数据查询

1. SELECT 基本结构

语句格式：

```
SELECT [ALL|DISTINCT]<目标列表达式>[,<目标列表达式> ]…
FROM <表名或视图名>[,<表名或视图名>]
[WHERE<条件表达式>]
[GROUP BY <列名 1>[HAVING<条件表达式>]]
[ORDER BY<列名 2>[ASC|DESC]];
```

SQL 查询中的子句顺序为 SELECT、FROM、WHERE、GROUP BY、HAVING 和 ORDER BY。SELECT、FROM 是必需的，HAVING 子句只能与 GROUP BY 搭配使用。

(1)　SELECT 子句对应的是关系代数中的投影运算，用来列出查询结果中的属性。其输出可以是列名、表达式、集函数(AVG、COUNT、MAX、MIN、SUM)，DISTINCT 选项可以保证查询的结果集中不存在重复元组。

(2) FROM 子句对应的是关系代数中的笛卡儿积，它列出的是表达式求值过程中须扫描的关系，即在 FROM 子句中出现多个基本表或视图时，系统首先执行笛卡儿积操作。

(3) WHERE 子句对应的是关系代数中的选择谓词。WHERE 子句的条件表达式中可以使用的运算符如表 7-3 所示。

表 7-3　WHERE 子句的条件表达式中可以使用的运算符

运算符		含义	运算符		含义
集合成员运算符	IN NOT IN	在集合中 不在集合中	算术运算符	> ⩾ < ⩽ = ≠	大于 大于等于 小于 小于等于 等于 不等于
字符串匹配运算符	LIKE	与 _ 和 % 进行字符匹配			
空值比较运算符	IS NULL IS NOT NULL	为空 不为空	逻辑运算符	AND OR NOT	与 或 非

2. 简单查询

SQL 最简单的查询是找出关系中满足特定条件的元组，这些查询与关系代数中的选择操作类似。简单查询只需要使用三个保留字 SELECT、FROM 和 WHERE。

3. 连接查询

若查询涉及两个以上的表，则称为连接查询。

4. 子查询与聚集函数

1) 子查询

子查询也称为嵌套查询，是指一个 SELECT-FROM-WHERE 查询可以嵌入另一个查询块之中。在 SQL 中允许多重嵌套。

2) 聚集函数

聚集函数是一个值的集合为输入，返回单个值的函数。SQL 提供了五个预定义集合函数：平均值 AVG、最小值 MIN、最大值 MAX、求和 SUM 以及计数 COUNT。

使用 ANY 和 ALL 谓词必须同时使用比较运算符，其含义及等价的转换关系如表 7-4 所示。

表 7-4　ANY 和 ALL 谓词的含义及等价的转换关系

谓词	语义	等价转换关系
>ANY	大于子查询结果中的某个值	>MIN
>ALL	大于子查询结果中的所有值	>MAX
<ANY	小于子查询结果中的某个值	<MAX

续表

谓　词	语　义	等价转换关系
<ALL	小于子查询结果中的所有值	<MIN
>=ANY	大于等于子查询结果中的某个值	>=MIN
>=ALL	大于等于子查询结果中的所有值	>=MAX
<=ANY	小于等于子查询结果中的某个值	<=MAX
<=ALL	小于等于子查询结果中的所有值	<=MIN
<>ANY	不等于子查询结果中的某个值	
<>ALL	不等于子查询结果中的所有值	NOT IN
=ANY	等于子查询结果中的某个值	IN
=ALL	等于子查询结果中的所有值	

5. 分组查询

1)　GROUP BY 子句

在 WHERE 子句后面加上 GROUP BY 子句可以对元组进行分组，保留字 GROUP BY 后面跟着一个分组属性列表。

2)　HAVING 子句

假如元组在分组前按照某种方式加上限制，使得不需要的分组为空，在 GROUP BY 子句后面跟一个 HAVING 子句即可。

当元组含有空值时，应该注意以下两点。

- 空值在任何聚集操作中都被忽略。它对求和、求平均值和计数都没有影响，也不能是某列的最大值或最小值。
- NULL 值可以在分组属性中看做一个一般的值。

6. 更名运算

SQL 提供了可为关系和属性重新命名的机制，这是通过使用具有如下形式的 AS 子句来实现的。

```
old-name AS new-name
```

AS 子句既可以出现在 SELECT 子句中，也可以出现在 FROM 子句中。

7. 字符串操作

对于字符串进行的最通常的操作是使用 LIKE 操作符的模式匹配。可以使用两个特殊的字符来描述模式：“%”匹配任意字符串；“-”匹配任意一个字符。

8. 视图查询

查询视图表时，系统先从数据字典中取出该视图的定义，然后将定义中的查询语句和对该视图的查询语句结合起来，形成一个修正的查询语句。

真题链接

【例 7-2】给定关系模式 R(A,B,C,D)、S(C,D,E)，与 $\pi_{1,3,5}(\sigma_{2=\text{'软件工程'}}=(R \bowtie S))$等价的 SQL 语句如下：

```
SELECT (53) FROM,S WHERE(54);
```

下列查询 B=“信息”且 E=“北京”的 A、B、E 的关系代数表达式中，查询效率最高的是 (55) 。(2014 年 11 月真题 53～55)

(53) A. A,C,S,C　　B. A,B,E　　C. A,R,C,E　　D. A,R,C,S,D

(54) A. B＝软件工程 OR R.C=S.C AND R.D=S.D

B. B= ‘软件工程’ OR R.C=S.C AND R.D=S.D

C. B= ‘软件工程’ OR R.C=S.C OR R.D=S.D

D. B= ‘软件工程’ AND R.C=S.C AND R.D=S.D

(55) A. $\pi_{1,2,7}(\sigma_{2=\text{'信息'}\wedge 3=5\wedge 4=6\wedge 7=\text{'北京'}}(R\times S))$

B. $\pi_{1,2,7}(\sigma_{3==5\wedge 4=6}(\sigma_{2=\text{'信息'}}(R)\times\sigma_{5=\text{'北京'}}(S)))$

C. $\pi_{1,2,7}(\sigma_{3==5\wedge 4=6\wedge 2=\text{''}}(R\times\sigma_{7=\text{' '}}(S)))$

D. $\pi_{1,2,7}(\sigma_{3==5\wedge 4=6\wedge 7=\text{'北京'}}(\sigma_{2=\text{'信息'}}(R)\times S)))$

【解析】(55)中，B 的最长查询语句最短，因此可以并行执行，提高效率。

【答案】(53)C；(54)D；(55)B

【例 7-3】假定某企业 2014 年 5 月的员工工资如表 7-5 所示。(2014 年 5 月真题 54～56)

表 7-5　2014 年 5 月员工工资表

员工号	姓名	部门	基本工资	岗位工资	全勤奖	应发工资	扣款	实发工资
1001	王小龙	办公室	680.00	1200.00	100.00	1980.00	20.00	1960.00
1002	孙晓红	办公室	1200.00	1000.00	0.00	2200.00	50.00	2150.00
2001	赵昍珊	企划部	680.00	1200.00	100.00	1980.00	10.00	1970.00
2002	李丽敏	企划部	950.00	2000.00	100.00	3050.00	15.00	3035.00
3002	傅学君	设计部	800.00	1800.00	0.00	2600.00	50.00	2550.00
3003	曹海军	设计部	950.00	1600.00	100.00	2650.00	20.00	2630.00
3004	赵晓勇	设计部	1200.00	2500.00	0.00	3700.00	50.00	3650.00
4001	杨一凡	销售部	680.00	1000.00	100.00	1780.00	10.00	1770.00
4003	景昊星	销售部	1200.00	2200.00	100.00	3500.00	20.00	3480.00
4005	李建军	销售部	850.00	1800.00	100.00	2750.00	98.00	2652.00

查询人数大于 2 的部门和部门员工应发工资的平均工资的 SQL 语句如下：

```
SELECT (54)
FROM 工资表
  (55)
  (56) ;
```

(54) A. 部门，AVG(应发工资) AS 平均工资　　B. 姓名，AVG(应发工资) AS 平均工资

C. 部门，平均工资 AS AVG(应发工资)　　D. 姓名，平均工资 AS AVG(应发工资)

(55) A. ORDER BY 姓名　　B. ORDER BY 部门

C. GROUP BY 姓名　　D. GROUP BY 部门

(56) A. WHERE COUNT(姓名)>2　　B. WHERE COUNT(DISTINCT(部门))>2

C. HAVING COUNT(姓名)>2　　D. HAVING COUNT(DISTINCT(部门))>2

【解析】本题要查询的是“部门”和“部门员工应发工资的平均工资”，查询条件是“部门人数大于2”。选项中，将 AVG(应发工资)命名为“平均工资”，重命名用 AS 子句实现：Old-name As New-name，可见(54)应选 A。本题在查询中应按部门分组，并加上条件“部门人数大于 2”。分组应采用 GROUP BY，分组的依据是“部门”，对分组加上限制使用 HAVING 子句，COUNT(姓名)可统计各个部门中的人数，而 COUNT(部门)统计的则是部门数。因此，完整的 SQL 语句应是：

```
SELECT  部门, AVG(应发工资) AS 平均工资
FROM 工资表
GROUP BY 部门
HAVING COUNT(姓名)>2;
```

【答案】(54)A；(55)D；(56)C

7.4.4　SQL 数据更新

1. 插入

插入语句的基本格式：

```
INSERT INTO 基本表名 [(字段名[,字段名>]…)]
        VALUE(常量[,常量]…);查询语句
INSERT INTO 基本表名(列表名)
        SELECT 查询语句
```

2. 删除

语句格式：

```
DELETE FROM 基本表名
        [WHERE 条件表达式]
```

3. 修改

语句格式：

```
UPDATE 基本表名
        SET 列名=值表达式[,列名=值表达式…]
        [WHERE 条件表达式];
```

7.4.5　SQL 的访问控制

数据控制是控制用户的数据存储权利，是由 DBA 来决定的。DBMS 数据控制应具有如

下功能。

- 通过 GRANT 和 REVOKE 将授权通知系统，并存入数据字典。
- 当用户提出请求时，根据授权情况检查是否执行操作请求。

SQL 标准包括 DELETE、INSERT、SELECT 和 UPDATE 权限。

1. 授权的语句格式

授权的语句格式如下。

```
GRANT <权限>[,<权限>]…
      [ON<对象类型><对象名>]
      TO<用户>[,<用户>]…
[WITH GRANT OPTION];
```

不同类型的操作对象有不同的操作权限，常见的操作权限如表 7-6 所示。

表 7-6　常见的操作权限

对　象	对象类型	操作权限
属性列	TABLE	SELECT、INSERT、UPDATE、DELETE、ALL PRIVILEGES(4 种权限总和)
视图	TABLE	SELECT、INSERT、UPDATE、DELETE、ALL PRIVILEGES(4 种权限总和)
基本表	TABLE	SELECT、INSERT、UPDATE、DELETE、ALTER、INDEX、ALL PRIVILEGES(6 种权限总和)
数据库	DATABASE	CREATETAB 建立表的权限，可由 DBA 授予普通用户

说明如下。

(1) PUBLIC：接受权限的用户可以是单个或多个具体的用户，PUBLIC 参数可将权限赋给全体用户。

(2) WITH GRANT OPTION：若指定了此子句，获得权限的用户还可以将权限赋给其他用户。

2. 收回权限语句格式

收回权限的语句格式：

```
REVOKE<权限>[,<权限>]…[ON<对象类型><对象名>]…
    FROM<用户>[,<用户>]…,
```

7.4.6　嵌入式 SQL

SQL 提供了将 SQL 语句嵌入到某种高级语言中的使用方式，识别嵌入在高级语言中的 SQL 语句通常采用预编译的方法。该方法的关键问题是必须区分主语言中嵌入的 SQL 语句，以及主语言和 SQL 间的通信问题。采用的方法是由 DBMS 的预处理程序对源程序进行扫描，识别出 SQL 语句，把它们转换为主语言调用语句，以使主语言编译程序能识别它，最后由主语言的编译程序将整个源程序编译成目标码。

嵌入式 SQL 与主语言之间的通信采用三种方式。

(1) SQL 通信区：向主语言传递 SQL 语句执行的状态信息，使主语言能够根据此信息控制程序流程。

(2) 主变量：也称共享变量。主语言向 SQL 语句提供参数主要通过主变量，主变量由主语言的程序定义，并用 SQL 的 DECLARE 语句说明。

(3) 游标 SQL：语言是面向集合的，一条 SQL 语句可产生或处理多条记录。而主语言是面向记录的，一组主变量一次只能放一条记录，所以引入游标，通过移动游标指针来决定获取哪条记录。

7.5　关系数据库的规范化

考核说明： 本节主要介绍函数依赖、规范化和模式分解等内容，是重点也是难点，要能理解并掌握。

7.5.1　函数依赖

数据依赖是通过一个关系中属性间值的相等与否体现出来的数据间的相互关系，是现实世界属性间相互联系和约束的抽象，是数据内在的性质，是语义的体现。函数依赖则是一种最重要、最基本的数据依赖。

(1) 函数依赖：设 $R(U)$是一个属性集 U 上的关系模式，X 和 Y 是 U 的子集。若对 $R(U)$ 的任何一个可能的关系 r，r 中不可能存在两个元组在 X 上的属性值相等，而在 Y 上的属性值不等，则称 X 函数决定 Y 或 Y 函数依赖于 X，记作 $X \to Y$。

(2) 非平凡的函数依赖：如果 $X \to Y$，但 $Y \not\subseteq X$，则称 $X \to Y$ 是非平凡的函数依赖。

(3) 平凡的函数依赖：如果 $X \to Y$，但 $Y \subset X$，则称 $X \to Y$ 是平凡的函数依赖。

(4) 完全函数依赖：在 $R(U)$中，如果 $X \to Y$，并且对于 X 的任何一个真子集 X'，都有 X'不能决定 Y，则称 Y 对 X 完全函数依赖，记作 $X \xrightarrow{f} Y$。

(5) 部分函数依赖：如果 $X \to Y$，但 Y 不完全函数依赖于 X，则称 Y 对 X 部分函数依赖，记作 $X \xrightarrow{p} Y$。部分函数依赖也称局部函数依赖。

(6) 传递依赖：在 $R(U, F)$中，如果 $X \to Y$，$Y \not\subseteq X$，Y 不能函数决定 X，$Y \to Z$，则称 Z 对 X 传递依赖。

(7) 码：设 K 为 $R(U, F)$中的属性的组合，若 $K \to U$，且对于 K 的任何一个真子集 K'，都有 K'不能决定 U，则 K 为 R 的候选码，若有多个候选码，则选一个作为主码。候选码通常也称候选关键字。

(8) 主属性和非主属性：包含在任何一个候选码中的属性叫作主属性，否则叫作非主属性。

(9) 外码：若 $R(U)$中的属性或属性组 X 非 R 的码，但 X 是另一个关系的码，则称 X 为外码。

(10) 函数依赖的公理系统(Armstrong 公理系统)：设关系模式 $R(U, F)$中，U 为属性集，F 是 U 上的一组函数依赖，那么有如下的推理规则。

- A1 自反律(Reflexivity)：若 $Y \subseteq X \subseteq U$，则 $X \to Y$ 为 F 所蕴涵。
- A2 增广律(Augmentation)：若 $X \to Y$ 为 F 所蕴涵，且 $Z \subseteq U$，则 $XZ \to YZ$ 为 F 所蕴涵。
- A3 传递律(Transitivity)：若 $X \to Y$，$Y \to Z$ 为 F 所蕴涵，则 $X \to Z$ 为 F 所蕴涵。

根据以上三条推理规则可以推出下面三条推理规则。

- 合并规则：若 $X \to Y$，$X \to Z$，则 $X \to YZ$ 为 F 所蕴涵。
- 伪传递率：若 $X \to Y$，$WY \to Z$，则 $XW \to Z$ 为 F 所蕴涵。
- 分解规则：若 $X \to Y$ 及 $Z \subseteq Y$，则 $X \to Z$ 为 F 所蕴涵。

引理：$X \to A_1 A_2 \cdots A_k$ 成立的充分必要条件是 $X \to A_i\ (i=1,2,\cdots,k)$成立。

真题链接

【例 7-4】给定关系模式 R(U,F)，U=(A,B,C,D,E,H)，函数依赖集 F={A->B,A->C,C->D,AE->H}。关系模式 R 的候选关键字为_(56)_。(2014 年 11 月真题 56)

(56) A. AC　　B. AB　　C. AE　　D. DE

【解析】通过AE关键字可以关联A, B, C, D, E, H所有属性，其他选择均不可。

【答案】C

7.5.2 规范化

关系数据库设计的方法之一就是设计满足适当范式的模式，通常可以通过判断分解后的模式达到几范式来评价模式规范化的程度。范式有 1NF、2NF、3NF、BCNF、4NF 和 5NF，其中 1NF 级别最低。这几种范式之间 5NF⊂4NF⊂BCNF⊂3NF⊂2NF⊂1NF 成立。通过分解，可以将一个低一级范式的关系模式转换成若干个高一级范式的关系模式，这种过程叫作规范化。

1. 1NF(第一范式)

【定义 7-4】若关系模式 *R* 的每一个分量是不可再分的数据项，则关系模式 *R* 属于第一范式(1NF)。

1NF 存在下面四个问题。

- 冗余度大。
- 引起修改操作的不一致性。
- 插入异常。
- 删除异常。

2. 2NF(第二范式)

【定义 7-5】若关系模式 $R \in$ 1NF，且每一个非主属性完全依赖于码，则关系模式 $R \in$ 2NF。

换句话说，当 1NF 消除了非主属性对码的部分函数依赖，则称为 2NF。

3. 3NF(第三范式)

【定义 7-6】若关系模式 $R(U, F)$中不存在这样的码 X、属性组 Y 及非主属性 Z(Z 不属于

Y)，使得 $X\to Y$，$Y\nrightarrow X$，$Y\to Z$ 成立，则称关系模式 $R\in$3NF。即当 2NF 消除了非主属性对码的传递函数依赖，则称为 3NF。

4. BCNF(巴克斯范式)

【定义 7-7】若关系模式 $R\in$1NF，$X\to Y$，且 Y 属于 X，X 必含有码，则关系模式 $R\in$BCNF。即当 3NF 消除了主属性对码的部分和传递函数依赖，则称为 BCNF。

一个满足 BCNF 的关系模式，应具有如下性质。

- 所有非主属性对每一个码都是完全函数依赖。
- 所有非主属性对每一个不包含它的码，也是完全函数依赖。
- 没有任何属性完全函数依赖于非码的任何一组属性。

真题链接

【例 7-5】设有关系模式 R(E,N,M,L,Q)，其函数依赖集为 F={E→N,EM→Q,M→L}，则关系模式 R 达到了__(53)__，该关系模式__(54)__。(2012 年 11 月真题 53～54)

(53) A. 1NF　　B. 2NF　　C. 3NF　　D. BCNF

(54) A. 无须进行分解，因为已经达到了 3NF

B. 无须进行分解，因为已经达到了 BCNF

C. 尽管不存在部分函数依赖，但还存在传递依赖，所以需要进行分解

D. 需要进行分解，因为存在冗余、修改操作的不一致性、插入和异常删除

【解析】关系 R 中的每一个属性都是不可再分的，所以该关系符合 1NF，但不满足每一个非主属性完全依赖于码的条件，达不到 2NF。1NF 存在冗余、修改操作的不一致性、插入和异常删除等问题，需要对该关系模式进行分解。

【答案】(53)A；(54)D

7.5.3　模式分解及分解应具有的特性

1. 分解

【定义 7-8】关系模式 $R(U, F)$的一个分解是指，$\rho=\{R_1<U_1, F_1>, R_2<U_2, F_2>, \cdots, R_n<U_n, F_n>\}$，其中 $U=U_1\cup U_2\cup\cdots\cup U_n$，并且没有 $U_i\subseteq U_j(1\leqslant i, j\leqslant n)$，$F_i$ 为 F 在 U_i 上的投影，$F_i=\{X\to Y \mid X\to Y\in F^+\wedge XY\subseteq U_i\}$。

对一个给定的模式进行分解，使得分解后的模式是否与原来的模式等价有三种情况。

- 分解具有无损连接性。
- 分解要保持函数依赖。
- 分解既要有无损连接性，又要保持函数依赖。

2. 无损连接

【定义 7-9】 $\rho=\{R_1<U_1,F_1>,R_2<U_2,F_2>,\cdots,R_n<U_n,F_n>\}$是关系模式 $R<U,F>$的一个分解，若对 R 的任何一个关系 r 均有 $r=m_\rho(r)$ 成立，则称分解 ρ 具有无损连接性(简称无损分解)。

其中 $m_{\rho}(r)=\bowtie_{i=1}^{k}\pi_{R_i}(r)$。

定理：关系模式 $R<U,F>$的一个分解 $\rho=\{R_1<U_1,F_1>,R_2<U_2,F_2>\}$具有无损连接的充分必要条件是

$$U_1\cap U_2\to U_1-U_2\in F^{+}\text{或}U_1\cap U_2\to U_2-U_1\in F^{+}$$

3. **保持函数依赖**

【定义 7-10】设关系模式 $R<U,F>$的一个分解 $\rho=\{R_1<U_1,F_1>,R_2<U_2,F_2>,\cdots,R_k<U_k,F_k>\}$，如果 $F^{+}=\left(\bigcup_{i=1}^{k}\pi_{R_i}(F^{+})\right)$，则称分解 ρ 保持函数依赖。

7.6 数据库的控制功能

考核说明： 本节主要介绍事务管理、数据库的备份与恢复、并发控制等内容，其中数据库的备份较为重要。

7.6.1 事务管理

事务是一个操作序列，这些操作“要么都做，要么都不做”，是数据库环境中不可分割的逻辑工作单位。

事务具有原子性(atomicity)、一致性(consistency)、隔离性(isolation)和持久性(durability)。这 4 个特性也称事务的 ACID 性质。

真题链接

【例 7-6】“当多个事务并发执行时，任一事务的更新操作直到其成功提交的整个过程对其他事务都是不可见的”，这一性质通常被称为事务的 (53) 。(2014 年 5 月真题 53)

(53) A. 原子性　　B. 一致性　　C. 隔离性　　D. 持久性

【解析】事务具有原子性、一致性、隔离性和持久性。

事务的原子性表示事务执行过程中的任何失败都将导致事务所做的任何修改失效。

一致性表示当事务执行失败时，所有被该事务影响的数据都应该恢复到事务执行前的状态。

隔离性表示在事务执行过程中对数据的修改，在事务提交之前对其他事务不可见。

持久性表示已提交的数据在事务执行失败时，数据的状态都应该正确。

【答案】C

7.6.2 数据库的备份与恢复

数据库的关键技术在于建立冗余数据，即备份数据。如何在系统出现故障后能够及时地使数据库恢复到故障前的正确状态，就是数据库恢复技术。

数据库中的 4 类故障是事务内部故障、系统故障、介质故障及计算机病毒。

建立冗余数据的方法是进行数据转储和登记日志文件。数据的转储分为静态转储和动态转储、海量转储和增量转储。

- 静态转储是指在转储期间不允许对数据库进行任何存取、修改操作。
- 动态转储是在转储期间允许对数据库进行存取、修改操作。
- 海量转储是指每次转储全部数据。
- 增量转储是指每次只转储上次转储后更新过的数据。

在事务处理的过程中，DBMS 把事务开始、事务结束以及对数据库的插入、删除和修改的每一次操作都写入日志文件。DBMS 利用日志文件进行事务故障恢复和系统故障恢复，并可协助后备副本进行介质故障恢复。

为了避免磁盘介质出现故障影响数据库的可用性，许多 DBMS 提供数据库镜像功能用于数据库恢复。

真题链接

【例 7-7】为了保证数据库中数据的安全可靠和正确有效，系统在进行事务处理时，对数据的插入、删除或修改的全部有关内容先写入__(51)__；当系统正常运行时，按一定的时间间隔，把数据库缓冲区内容写入__(52)__；当发生故障时，根据现场数据内容及相关文件来恢复系统的状态。(2014 年 5 月真题 51～52)

(51) A. 索引文件　B. 数据文件　C. 日志文件　D. 数据字典

(52) A. 索引文件　B. 数据文件　C. 日志文件　D. 数据字典

【解析】数据库管理系统利用日志文件进行事务故障恢复和系统故障恢复。在事务处理过程中，DBMS 把事务开始、事务结束以及对数据库的插入、删除和修改的每一次操作写入日志文件。当系统正常运行时，按一定的时间间隔，把数据库缓冲区内容写入数据文件；一旦发生故障，DBMS 的恢复子系统利用日志文件撤销事务对数据库的改变，回退到事务的初始状态。

【答案】(51)C；(52)B

7.6.3　并发控制

所谓并发操作，是指在多用户共享的系统中许多用户可能同时对同一数据进行操作。并发操作带来的问题是数据的不一致性，主要有三类：丢失更新、不可重复读和读脏数据。其主要原因是事务的并发操作破坏了事务的隔离性。DBMS 的并发控制子系统负责协调并发事务的执行，保证数据库的完整性不受破坏，避免用户得到不正确的数据。

并发控制的主要技术是封锁。基本封锁的类型有排它锁(简称 X 锁或写锁)和共享锁(简称 S 锁或读锁)。

7.7　数据库的分析与设计

考核说明：本节主要介绍数据库分析和数据设计的步骤，是下午考试必考内容，要重点掌握数据库概念设计模型和关系模式设计。

7.7.1 数据库设计的步骤

按照软件工程对系统生命周期的定义，软件生命周期分为六个阶段工作：制订计划、需求分析、设计、程序编制、测试及运行维护。在数据库设计中也参照这种划分，把数据库应用系统的生命周期分为数据库规划、需求描述与分析、数据库与应用程序设计、数据库系统实现、测试和运行维护六个阶段。

(1) 数据库规划。数据库规划是创建数据库应用系统的起点，是数据库应用系统的任务陈述和任务目标。

(2) 需求描述与分析。需求描述与分析是站在用户的角度，从系统中的数据和业务规则入手，收集和整理用户的信息，以特定的方式加以描述。它是下一步工作的基础。

(3) 数据库与应用程序设计。数据库设计是对用户数据的组织和存储设计；应用程序设计是在数据库设计的基础上对数据操作及业务实现的设计，包括事务设计和用户界面设计。

(4) 数据库系统实现。数据库系统实现是依照设计，使用 DBMS 支持的数据定义语言(DDL)实现数据库的建立，用高级语言(Basic、Delphi、C、C++和 PowerBuilder 等)编写应用程序。

(5) 测试。测试阶段是在数据系统投入使用之前，通过精心制订的测试计划和测试数据来测试系统的性能是否满足设计要求，并发现问题。

(6) 运行维护。数据库应用系统经过测试、试运行后即可正式投入运行。系统投入使用后，必须不断地对其进行评价、调整与修改等运行维护，直至系统消亡。

7.7.2 数据库设计的方法

在确定了数据库设计的策略以后，还需要选择相应的设计方法和步骤。多年来，人们提出了多种数据库设计方法、设计准则和规范。

1978 年 10 月召开的新奥尔良(New Orleans)会议提出的关于数据库设计的步骤(简称新奥尔良法)是目前得到公认的、较完整的、较权威的数据库设计方法，它把数据库设计分为以下四个主要阶段。

(1) 用户需求分析。用户需求分析是数据库设计人员采用一定的辅助工具对应用对象的功能、性能和限制等要求所进行的科学分析。

(2) 概念设计。概念设计是对信息进行分析和定义，如视图模型化、视图分析和汇总。该阶段对应用对象精确地进行抽象和概括，以形成独立于计算机系统的企业信息模型。描述概念模型较理想的工具是 E-R 图。

(3) 逻辑设计。逻辑设计是将抽象的概念模型转化为与选用的 DBMS 产品所支持的数据模型相符合的逻辑模型，它是物理设计的基础，包括模式初始设计、子模式设计、应用程序设计、模式评价及模式求精。

(4) 物理设计。物理设计是逻辑模型在计算机中的具体实现方案。

当各阶段发现不能满足用户需求时，均需返回到前面适当的阶段进行必要的修正。如

此经过不断的迭代和求精，直到各种性能均能满足用户的需求为止。

真题链接

【例 7-8】在数据库逻辑结构设计阶段，需要__(51)__阶段形成的__(52)__作为设计依据。(2014 年 11 月真题 51～52)

(51) A. 需求分析　　B. 概念结构设计　　C. 物理结构设计　　D. 数据库运行和维护

(52) A. 程序文档、数据字典和数据流图　　B. 需求说明文档、程序文档和数据流图　　C. 需求说明文档、数据字典和数据流图　　D. 需求说明文档、数据字典和程序文档

【解析】按照规范的设计方法，数据库的设计过程一般分为以下六个阶段：

(1) 需求分析：分析用户的需求，包括数据、功能和性能需求；

(2) 概念结构设计：主要采用 E-R 模型进行设计，包括画 E-R 图；

(3) 逻辑结构设计：通过将 E-R 图转换成表，实现从 E-R 模型到关系模型的转换；

(4) 数据库物理设计：主要是为所设计的数据库选择合适的存储结构和存取路径；

(5) 数据库的实施：包括编程、测试和试运行；

(6) 数据库运行与维护：系统的运行与数据库的日常维护。

【答案】(51)A；(52)C

【例 7-9】阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。(2014 年 11 月下午试题二)

【说明】

某集团公司在全国不同城市拥有多个大型超市，为了有效管理各个超市的业务工作，需要构建一个超市信息管理系统。

【需求分析结果】

(1) 超市信息包括：超市名称、地址、经理和电话，其中超市名称唯一确定超市关系的每一个元组。每个超市只有一名经理。

(2) 超市设有计划部、财务部、销售部等多个部门，每个部门只有一名部门经理，有多名员工，每个员工只属于一个部门。部门信息包括：超市名称、部门名称、部门经理和联系电话。超市名称、部门名称唯一确定部门关系的每一个元组。

(3) 员工信息包括：员工号、姓名、超市名称、部门名称、职位、联系方式和工资。其中，职位信息包括：经理、部门经理、业务员等。员工号唯一确定员工关系的每一个元组。

(4) 商品信息包括：商品号、商品名称、型号、单价和数量。商品号唯一确定商品关系的每一个元组。一名业务员可以负责超市内多种商品的配给，一种商品可以由多名业务员配给。

【概念模型设计】

根据需求分析阶段收集的信息，设计的实体联系图(见图 7-2)和关系模式(不完整) 如下：

【关系模式设计】

超市(超市名称，经理，地址，电话)

部门(__(a)__，部门经理，联系电话)

员工(__(b)__，姓名，联系方式，职位，工资)

商品(商品号，商品名称，型号，单价，数量)

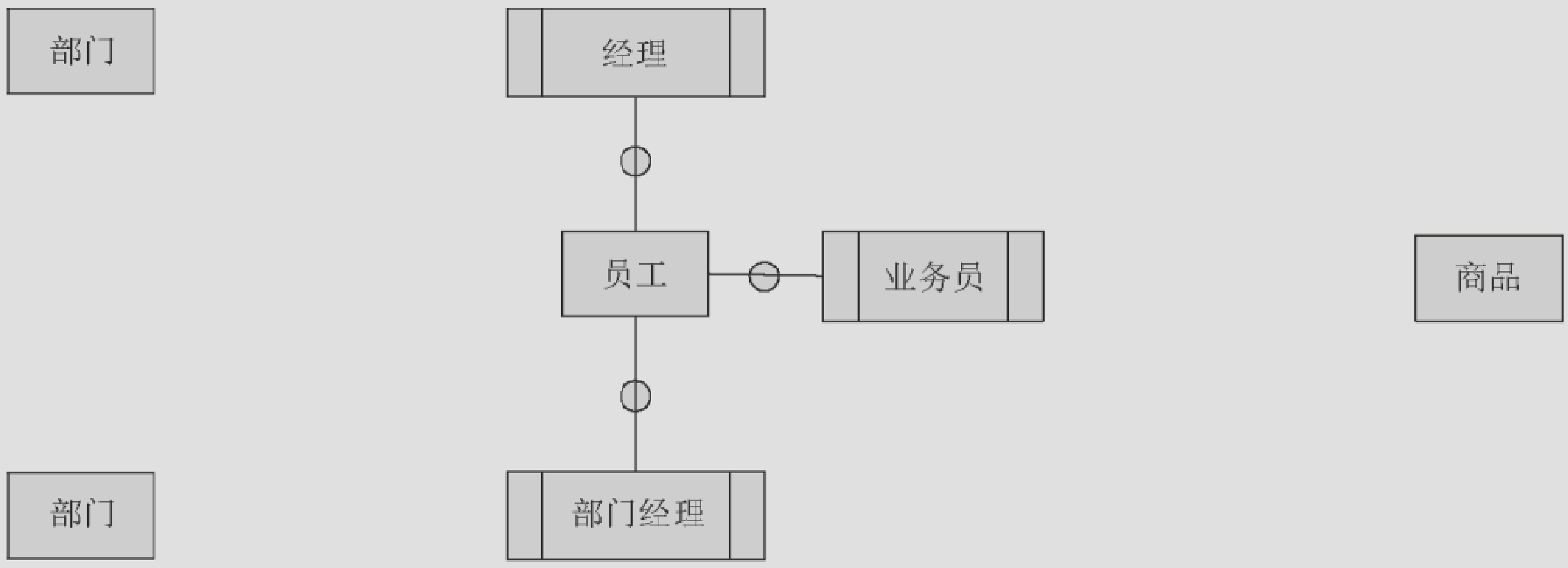


图 7-2　实体联系图

配给((c)，配给时间，配给数量，业务员)

【问题 1】

根据问题描述，补充四个联系，完善图 7-2 的实体联系图。联系名可用联系 1、联系 2、联系 3 和联系 4 代替，联系的类型分为 1:1、1:*n* 和 *m*:*n*(或 1:1、1:＊和＊:*)。

【问题 2】

(1) 根据实体联系图，将关系模式中的空(a)～(c)补充完整；

(2) 给出部门和配给关系模式的主键和外键。

【问题 3】

(1) 超市关系的地址可以进一步分为邮编、省、市、街道，那么该属性是属于简单属性还是复合属性？请用 100 字以内文字说明。

(2) 假设超市需要增设一个经理的职位，那么超市与经理之间的联系类型应修改为(d)，超市关系应修改为(e)。

【解析】

本题考查数据库系统中实体联系模型和关系模式设计方面的应用知识，属于比较传统的题目，考察点也和往年类似。

【问题 1】

本题考查数据库的概念结构设计。两个实体集间的联系分为三类：一对一(1:1)、一对多(1:*n*)和多对多(*m*:*n*)。根据题意，“每个部门只有一名部门经理”部门和部门经理之间是 1:1 关系，每个部门有多名员工，即部门和员工之间是 1:*n* 的关系，由“超市只有一名经理”可知表示超市的部门和经理之间是 1:1 的关系，由“超市有多个部门”可知超市和部门之间是 1:*n* 的关系。由“一名业务员可以负责超市内多种商品的配给，一种商品可以由多名业务员配给”可知业务员和商品之间是多对多的关系，即 *m*:*n*。根据以上分析，即可完成图 7-2 的实体联系图。

【问题 2】

本题考查数据库的逻辑结构设计，题目要求补充完整各关系模式，并给出部门和配给关系模式的主键和外键。根据问题一画完整的实体联系图和需求描述，员工关系模式包括员工号、姓名、部门名称、超市名称、职位、联系方式、工资等，因此(b)处应填员工号、超市名称和部门名称；部门关系模式包括超市名称、部门名称、部门经理和联系电话等，由此可知(a)处填超市名称和部门名称；配给关系模式包括商品号、

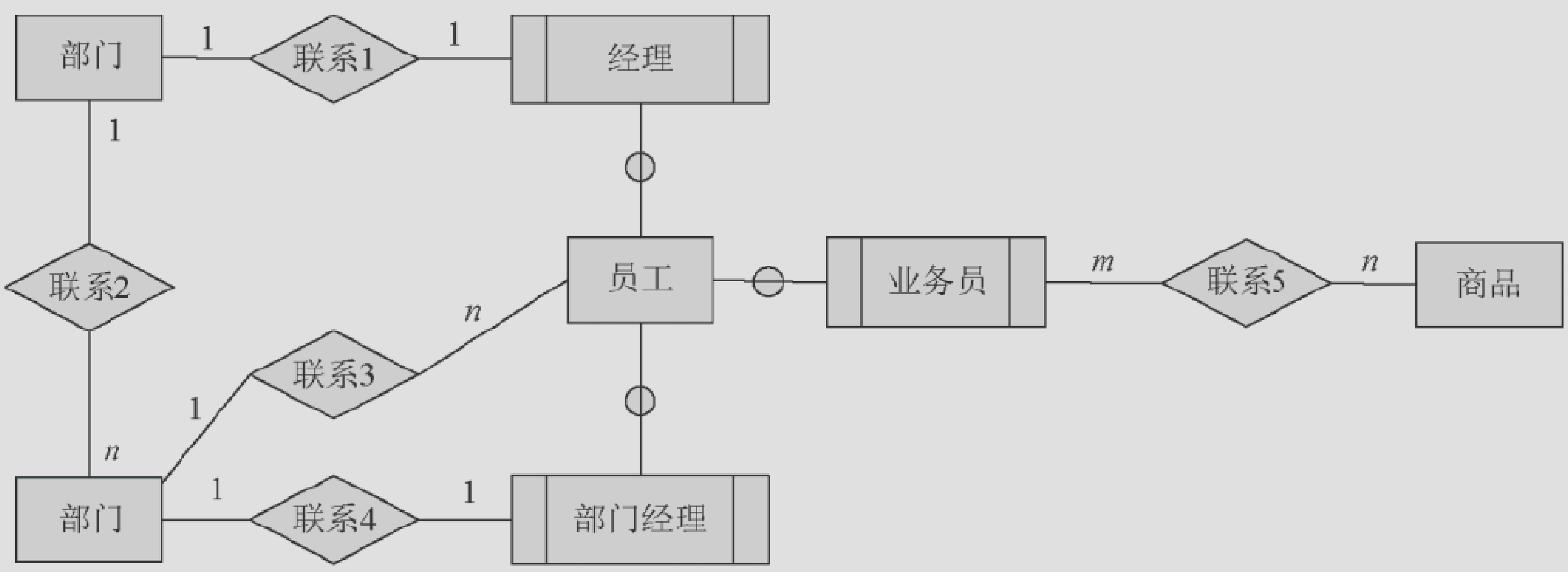
业务员、配给时间、配给数量、业务员等，由此可知，(c)处应填商品号、业务员。部门关系模式的主键应为(超市名称和部门名称)，外键名称为(超市名称、部门经理)；配给关系模式的主键应为(商品号、业务员、配给时间)，外键应为(业务员、商品号)。

【问题 3】

超市关系的地址可以进一步分为邮编、省、市、街道，则超市关系中的地址属于复合属性。所谓复合属性就是指属性中含有多种信息，可以进一步拆分的属性，地址可以拆分成多个简单属性，符合这一特征。假设超市需要增设一个经理的职位，那么超市与经理之间的联系类型应该是一个超市有多个经理，为 1:n 的关系，超市关系应包含超市名称、地址、电话。

【答案】

【问题 1】



【问题 2】

(a) 超市名称，部门名称；主键：(超市名称，部门名称)；外键：超市名称、部门经理

(b) 员工号，超市名称，部门名称

(c) 商品号；主键(商品号，业务员，配给时间)；外键：业务员、商品号

【问题 3】

(1) 超市关系中的地址属于复合属性。所谓复合属性就是指属性中含有多种信息，可以进一步拆分的属性，地址可以拆分成多个简单属性，符合这一特征。

(2) (d)1:n　　(e)超市名称、地址、电话

【例 7-10】阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。(2014 年 5 月下午试题二)

【说明】

某家电销售电子商务公司拟开发一套信息管理系统，以方便对公司的员工、家电销售、家电厂商和客户等进行管理。

【需求分析】

(1) 系统需要维护电子商务公司的员工信息、客户信息、家电信息和家电厂商信息等，员工信息主要包括工号、姓名、性别、岗位、身份证号、电话、住址，其中岗位包括部门经理和客服等，客户信息主要包括客户 ID、姓名、身份证号、电话、住址、账户余额。家电信息主要包括家电条码、家电名称、价格、出厂日期、所属厂商，家电厂商信息包括厂商 ID、厂商名称、电话、法人代表信息、厂址。

(2) 电子商务公司根据销售情况，由部门经理向家电厂商订购各类家电，每个家电厂商只能由一名部

门经理负责。

(3) 客户通过浏览电子商务公司网站查询家电信息，与客服沟通获得优惠后，在线购买。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图(不完整)如图 7-3 所示。

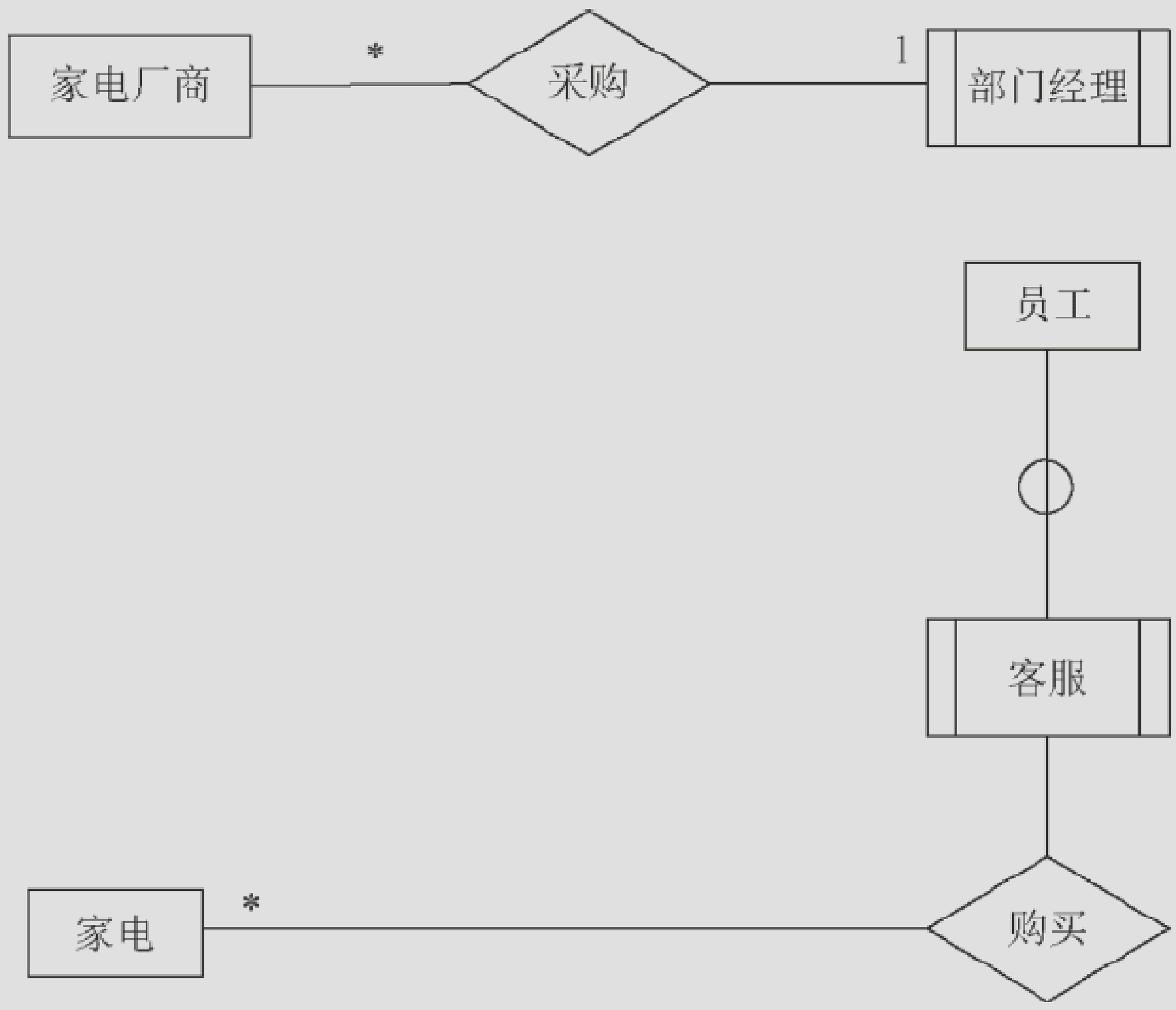


图 7-3 实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式(不完整)：

客户(客户 ID，姓名，身份证号，电话，住址，账户余额)

员工(工号，姓名，性别，岗位，身份证号，电话，住址)

家电(家电条码，家电名称，价格，出厂日期，(1))

家电厂商(厂商 ID，厂商名称，电话，法人代表信息，厂址，(2))

购买(订购单号，(3)，金额)

【问题 1】

补充图 7-3 中的联系和联系的类型。

【问题 2】

根据图 7-3，将逻辑结构设计阶段生成的关系模式中的空(1)～(3)补充完整，用下划线指出“家电”、“家电厂商”和“购买”关系模式的主键。

【问题 3】

电子商务公司的主营业务是销售各类家电，对账户有余额的客户，还可以联合第二方基金公司提供理财服务，为此设立客户经理岗位。客户通过电子商务公司的客户经理和基金公司的基金经理进行理财，每名客户只有一名客户经理和一名基金经理负责，客户经理和基金经理均可负责多名客户。请根据该要求，对图 7-3 进行修改，画出修改后的实体间联系和联系的类型。

解析：本题考查数据库设计。设计考点有数据库的概念结构设计和逻辑结构设计。

【问题 1】根据题意，每个家电厂商可以购买多种家电，因此家电厂商和家电之间是一对多的联系。一种家电可由多个客户购买，一个客户可购买多件家电，因此家电和客户之间是多对多的联系，由此可画

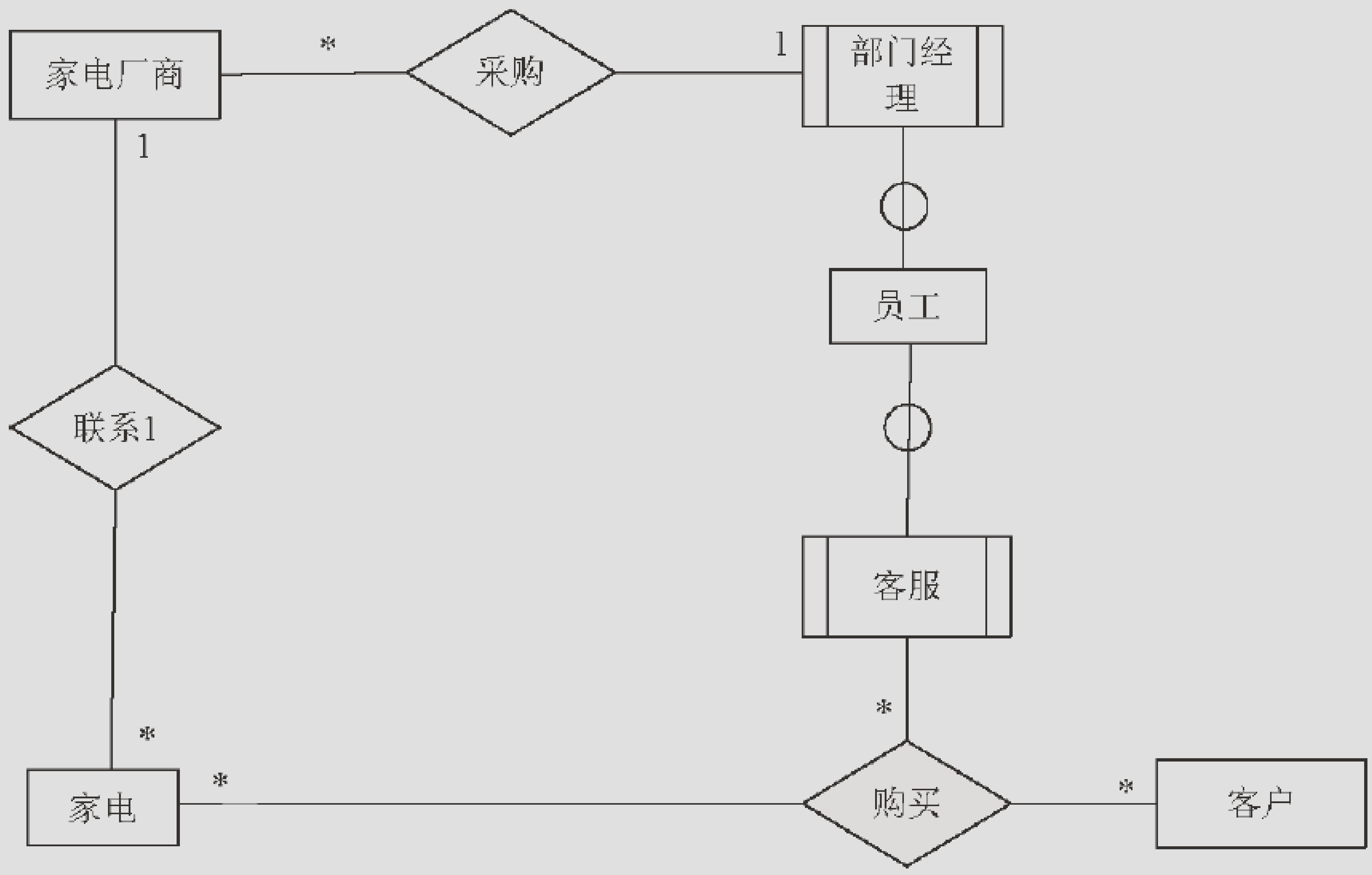
出完整的实体联系图。

【问题 2】电子商务公司根据销售情况，由部门经理向家电厂商订购各类家电，每个家电厂商只能由一名部门经理负责，家电厂商中需要保存是由哪名部门经理负责的，因此这里需要有员工的工号，这样才能在员工中找到唯一对应的员工。

【问题 3】由题意可知，可增设实体客户经理和基金经理，客户和客户经理、基金经理都有联系，且客户经理和客户、基金经理和客户都是一对多的联系，由此可补充图 7-3。

【答案】

【问题 1】



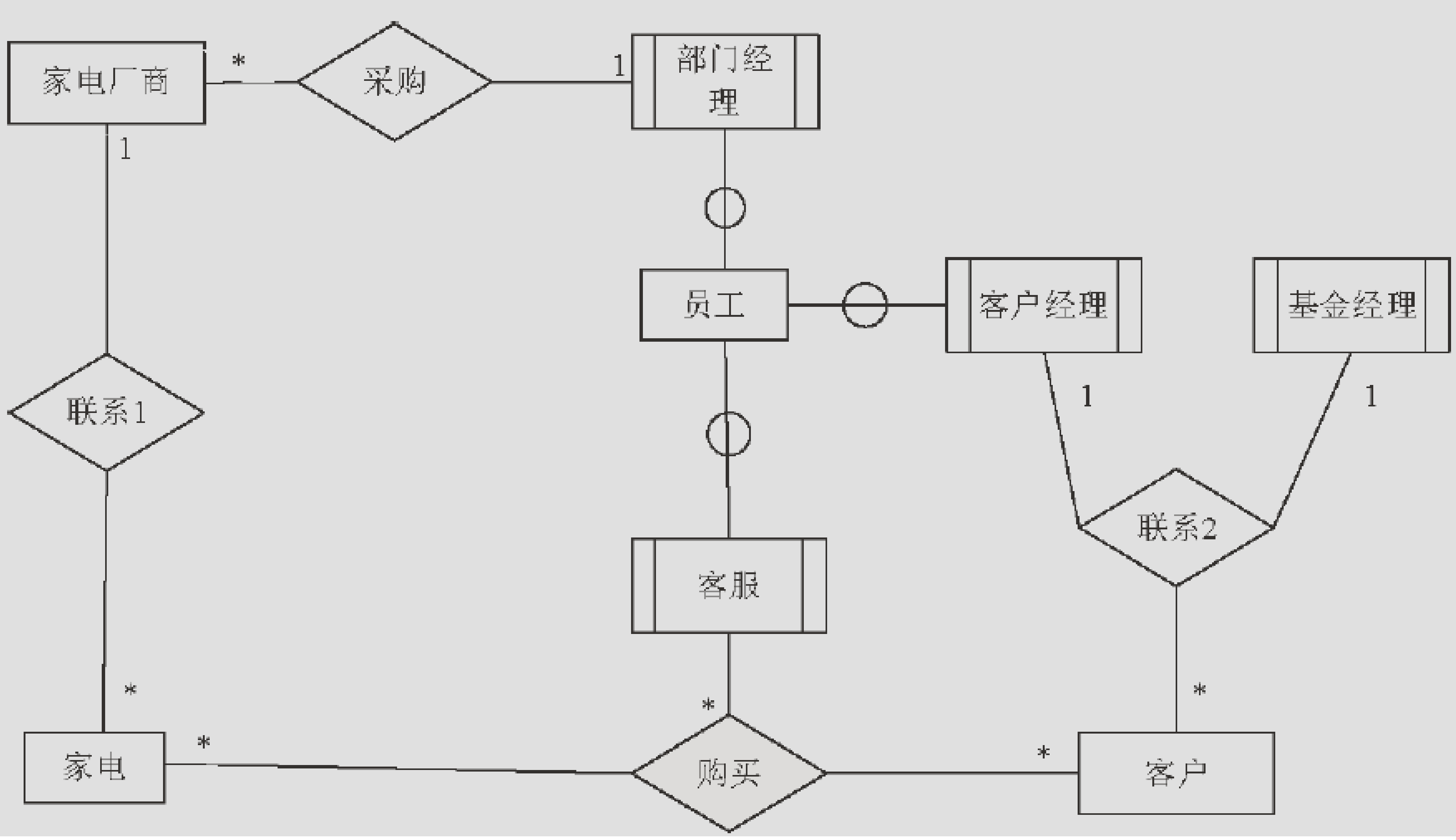
【问题 2】

家电(<u>家电条码</u>，家电名称，价格，出厂日期，厂商 ID)

家电厂商(<u>厂商 ID</u>，厂商名称，电话，法人代表信息，厂址，工号)

购买(<u>订购单号</u>，家电条码，客户 ID，工号，金额)

【问题 3】



7.8 应试加油站

7.8.1 考频统计

通过对近几年考试真题的分析，“数据库技术”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 7-7 所示。

表 7-7 历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：51～56	数据库设计过程、关系模式、SQL 语句	6
	下午：试题二	数据库分析与设计	15
2014 年 5 月	上午：51～56	数据库管理系统、事务管理、SQL 查询语句	6
	下午：试题二	数据库分析与设计	15
2013 年 11 月	上午：51～56	自然连接运算和 SQL 语句、分布式数据库系统、数据库备份	6
	下午：试题二	数据库分析与设计	15
2013 年 5 月	上午：54～59	视图的概念、函数依赖、SELECT 语句	6
	下午：试题二	数据库分析与设计	15
2012 年 11 月	上午：51～56	自然连接运算、规范化、创建视图的 SQL 语句	6
	下午：试题二	数据库分析与设计	15
2012 年 5 月	上午：49，52～56	E-R 模型、投影运算、1NF、SELECT 语句	6
	下午：试题二	数据库分析与设计	15

纵观历年试卷，本章知识点是以选择题和综合分析题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量大约为 6 道选择题，所占分值为 6 分(约占试卷总分值 75 分中的 8%)；在下午试卷中，所考查的题量为 1 道综合分析题，所占分值为 15 分(约占试卷总分值 75 分中的 20%)。本章试题偏重于实践应用，检验考生是否理解相关的理论知识点和实践经验，考试难度中等偏上。数据库设计是下午考试必考的内容，要重点掌握，尤其是 E-R 图和关系模式。

7.8.2 解题技巧

【典型题 7-1】在分布式数据库系统中，__(55)__是指用户无须知道数据存放的物理位置。(2013 年 11 月真题 55)

(55) A. 分片透明　　B. 复制透明　　C. 逻辑透明　　D. 位置透明

【解析】分片透明是最高层次的分布透明性，是指用户或应用程序只对全局关系进行操作而不必考虑数据的分片；复制透明性：在分布式系统中，为了提高系统的性能和实用性，有些数据并不只存放在一个

场地，很可能同时重复地存放在不同的场地；位置透明性：位于分片视图与分配视图之间。是指用户或应用程序应当了解分片情况，但不必了解片段的存储场地。所以本题答案选 D。

【答案】D

【典型题 7-2】计算机系统的软硬件故障可能会造成数据库中的数据被破坏。为了防止这一问题，通常需要__(56)__，以便发生故障时恢复数据库。(2013 年 11 月真题 56)

(56) A. 定期安装 DBMS 和应用程序

B. 定期安装应用程序，并将数据库做镜像

C. 定期安装 DBMS，并将数据库作备份

D. 定期将数据库作备份；在进行事务处理时，需要将数据更新写入日志文件

【解析】为了防止计算机系统的软硬件故障可能会造成数据库中的数据被破坏，通常要定期将数据库作备份；在进行事务处理时，需要将数据更新写入日志文件。

【答案】D

【典型题 7-3】若对关系 R(A，B，C，D)进行$\pi_{1,3}(R)$运算，则该关系运算与__(52)__等价，表示__(53)__。(2012 年 5 月真题 52～53)

(52) A. $\pi_{A=1,C=3}(R)$　　B. $\pi_{A=1 \wedge C=3}(R)$　　C. $\pi_{A,C}(R)$　　D. $\pi_{A=1 \vee C=3}(R)$

(53) A. 属性 A 和 C 的值分别等于 1 和 3 的元组为结果集

B. 属性 A 和 C 的值分别等于 1 和 3 的两列为结果集

C. 对 R 关系进行 A=1、C=3 的投影运算

D. 对 R 关系进行属性 A 和 C 的投影运算

【解析】$\pi_{1,3}(R)$表示对 R 进行投影运算，选择的是 R 的第 1 列和第 3 列，也就是 A 列和 C 列。

【答案】(52)C；(53)D

【典型题 7-4】E-R 模型向关系模型转换时，三个实体之间多对多的联系 m:n:p 应该转换为一个独立的关系模式，且该关系模式的关键字由__(49)__组成。(2012 年 5 月真题 49)

(49) A. 多对多联系的属性　　B. 三个实体的关键字

C. 任意一个实体的关键字　　D. 任意两个实体的关键字

【解析】一个m:n联系转换为一个关系模式时，与该联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。三个或三个以上实体间的一个多元联系转换为一个关系模式时，与该多元联系相连的各实体的码以及联系本身的属性均转换为关系的属性，而关系的码为各实体码的组合。

【答案】B

【典型题 7-5】在数据库系统中，视图是一个__(54)__。(2013 年 5 月真题 54)

(54) A. 真实存在的表，并保存了待查询的数据

B. 真实存在的表，只有部分数据来源于基本表

C. 虚拟表，查询时只能从一个基本表中导出

D. 虚拟表，查询时可以从一个或者多个基本表或视图中导出

【解析】视图是从一个或几个基本表(或视图)导出的虚拟表，其内容由查询定义。同真实的表一样，视图包含一系列带有名称的列和行数据。但是，视图并不在数据库中以存储的数据值集形式存在。行和列数据来自由定义视图的查询所引用的表，并且在引用视图时动态生成。

【答案】D

【典型题 7-6】(2013 年 5 月真题 56～59)

假设学生 Students 和教师 Teachers 的关系模式如下所示：

Students(学号，姓名，性别，类别，身份证号)

Teachers(教师号，姓名，性别，身份证号)

其中，学生关系中的类别分为“本科生”和“研究生”两类。

a. 查询在读研究生的教师的平均工资，最高与最低工资的差值的 SQL 语句如下：

```
SELECT ____(56)____
FROM Teachers , Students
WHERE ____(57)____;
```

(56) A. AVG(工资) AS 平均工资，MAX(工资)–MIN(工资) AS 差值

B. 平均工资 AS AVG(工资)，差值 AS MAX(工资)–MIN(工资)

C. AVG(工资)ANY 平均工资，MAX(工资)–MIN(工资)ANY 差值

D. 平均工资 ANY AVG(工资)，差值 ANY MAX(工资)–MIN(工资)

(57) A. Students.身份证号=Teachers.身份证号

B. Students.类别=‘研究生’

C. Students.身份证号=Teachers.身份证号 AND Students.类别=‘研究生’

D. Students.身份证号=Teachers.身份证号 OR Students.类别=‘研究生’

b. 查询既是女教师又是研究生且工资大于等于 3500 元的身份证号和姓名的 SQL 语句如下：

```
(SELECT 身份证号，姓名
  FROM Students
  WHERE____(58)____)
 INTERSECT (SELECT 身份证号，姓名
            FROM Teachers
            WHERE____(59)____)
```

(58) A. 工资>=3500

B. 工资>=‘3500’

C. 性别=‘女’AND 类别=研究生

D. 性别=‘女’AND 类别=‘研究生’

(59) A. 工资>=3500

B. 工资>=‘3500’

C. 性别=女 AND 类别=研究生

D. 性别=‘女’AND 类别=‘研究生’

【解析】查询在读研究生的教师的平均工资，最高与最低工资的差值的 SQL 语句为：SElECT AVG(工资) AS 平均工资，MAX(工资)–MIN(工资)AS 差值 FROM Teachers , Students WHERE Students.身份证号=Teachers.身份证号 AND Students.类别=‘研究生’。查询既是女教师又是研究生且工资大于等于 3500 元的身份证号和姓名的 SQL 语句：(SELECT 身份证号，姓名 FROM Students WHERE 性别=‘女’AND 类别=‘研究生’)INTERSECT (SELECT 身份证号，姓名 FROM Teachers WHERE 工资>=3500)。由此可知答案。

【答案】(56)A；(57)C；(58)D；(59)A

【典型题 7-7】(2013 年 11 月真题 51～54)

若有关系 R(A,B,C,D,E)和 S(B,C,F,G)，则 R 与 S 自然连接运算后的属性列有__(51)__个，与表达式 $\pi_{1,3,6,7}(\sigma_{3<6}(R \bowtie S))$ 等价的 SQL 语句如下。

```
SELECT  (52)   FROM  (53)   WHERE  (54)
```

(51) A. 5　　B. 6　　C. 7　　D. 9

(52) A. A,R.C,F,G　　B. A,C,S.B,S.F

C. A,C,S.B,S.C　　D. R.A,R.C,S.B,S.C

(53) A. R　　B. S　　C. RS　　D. R,S

(54) A. R.B=S.B AND R.C=S.C AND R.C<S.B

B. R.B=S.B AND R.C=S.C AND R.C<S.F

C. R.B=S.B OR R.C=S.C OR R.C<S.B

D. R.B=S.B OR R.C=S.C OR R.C<S.F

【解析】自然连接就是在等值连接的情况下，当连接属性 R 与 S 具有相同属性组时，把在连接结果中重复的属性列去掉，所以属性列有 7 个。

【答案】(51)C；(52)A；(53)D；(54)B

【典型题 7-8】(2012 年 11 月真题 55～56)

已知关系模式：图书(图书编号，图书类型，图书名称，作者，出版社，出版日期，ISBN)，图书编号唯一识别一本图书。建立“计算机”类图书的视图 Computer-BOOK，并要求进行修改、插入操作时保证该视图只有计算机类的图书。实现上述要求的 SQL 语句如下：

```
CREATE  (55)
AS SELECT 图书编号，图书名称，作者，出版社，出版日期
FROM 图书
WHERE 图书类型='计算机'
 (56) ;
(55)A. TABLE Computer-BOOK       B. VIEW Computer-BOOK
    C. Computer-BOOK TABLE       D. Computer-BOOK VIEW
(56)A. FOR ALL                   B. PUBLIC
    C. WITH CHECK OPTION         D. WITH GRANT OPTION
```

【解析】创建视图的SQL语句格式：

```
CREATE VIEW 视图名
    AS SELECT 查询子句
    [WITH CHECK OPTION];
```

其中，WITH CHECK OPTION 表示对 UPDATE、INSERT 和 DELETE 操作时保持更新、插入或删除的行满足视图定义中的谓词条件(即子查询中的条件表达式)。

【答案】(55)B；(56)C

【典型题 7-9】(2012 年 5 月真题 54～56)

某销售公司数据库的零件关系 P(零件号，零件名称，供应商，供应商所在地，库存量)，函数依赖集 F={零件号→零件名称，(零件号，供应商)→库存量，供应商→供应商所在地}。零件关系模式 P 属于__(54)__。

查询各种零件的平均库存量、最多库存量与最少库存量之间差值的 SQL 语句如下：

```
SELECT 零件号，零件名称， __(55)__ ，
FROM P
__(56)__
```

(54) A. 1NF　　B. 2NF　　C. 3NF　　D. 4NF

(55) A. AVG(库存量) AS 平均库存量，MAX(库存量)-MIN(库存量) AS 差值

B. 平均库存量 AS AVG(库存量)，差值 AS MAX(库存量)-MIN(库存量)

C. AVG 库存量 AS 平均库存量，MAX 库存量-MIN 库存量 AS 差值

D. 平均库存量 AS AVG 库存量，差值 AS MAX 库存量-MIN 库存量

(56) A. ORDER BY 供应商　　B. ORDER BY 零件号

C. GROUP BY 供应商　　D. GROUP BY 零件号

【解析】

1NF：关系模式 P 的每一个分量都是不可再分的数据项。

2NF：关系模式 P 属于 1NF，且每个非主属性完全依赖于码。

本题中，关系模式 P 的每个分量都不可以再分，满足 1NF 的定义。从函数依赖集 F 可以看出关系 P 的码为(零件号，供应商)，零件号决定零件名称，则零件名称不完全依赖于码，不满足 2NF 的定义。因此关系模式 P 属于 1NF。

查询平均库存需要使用 AVG()函数。计算最大值和最小值，则需要使用 MAX()和 MIN()函数。SELECT 语句可以通过 AS 子句为属性重新命名，形式为 old-name AS new-name，也就是说，新名称要放在 AS 的后面。

ORDER BY 子句用于排序，GROPU BY 子句用于分组。很显然，本题要按零件进行分组。

【答案】(54)A；(55)A；(56)D

【典型题 7-10】给定关系模式 R(U，F)，其中：属性集 U={A,B,C,D,E,G}，函数依赖集 F={A→B，A→C，C→D，AE→G}。若将 R 分解为 __(55)__ 两个模式，则分解后的关系模式保持函数依赖。(2013 年 5 月真题 55)

(55) A. R1(A,B,C)和 R2(D,E,G)　　B. R1(A,B,C,D)和 R2(A,E,G)

C. R1(B,C,D)和 R2(A,E,G)　　D. R1(B,C,D,E)和 R2(A,E,G)

【解析】本题考查数据库中函数依赖，函数依赖的定义：设 R(U)是属性集 U={A1, A2, …, An}上的关系模式，X 和 Y 是 U 的子集。若对 R(U)的任一具体关系 r 中的任意两个元组 t1 和 t2，只要 t1[X]=t2[X]就有 t1[Y]=t2[Y]。则称“X 函数确定 Y”或“Y 函数依赖于 X”(Founctional Dependence)，记作 X→Y。只有选项 B 中的两个关系式满足定义。

【答案】B

【典型题 7-11】(2013 年 11 月下午试题二)

阅读以下说明，根据要求回答问题 1～问题 3。

【说明】

某快递公司为了方便管理公司物品运送的各项业务活动，需要构建一个物品运送信息管理系统。

【需求分析结果】

(1) 快递公司有多个分公司，分公司信息包括分公司编号、名称、经理、办公电话和地址。每个分公

司可以有多名员工处理分公司的日常业务，每名员工只能在一个分公司工作。每个分公司由一名经理负责管理分公司的业务和员工，系统需要记录每个经理的任职时间。

(2) 员工信息包括员工号、姓名、岗位、薪资、手机号和家庭地址。其中，员工号唯一标识员工信息的每一个元组。岗位包括经理、调度员、业务员等。业务员根据客户提交的快件申请单进行快件受理事宜，一个业务员可以受理多个客户的快件申请，一个快件申请只能由一个业务员受理。调度员根据已受理的申请单安排快件的承运事宜，例如：执行承运的业务员、运达时间等。一个业务员可以执行调度员安排的多个快件的承运业务。

(3) 客户信息包括客户号、单位名称、通信地址、所属省份、联系人、联系电话、银行账号。其中，客户号唯一标识客户信息的每一个元组。当客户要寄快件时，先要提交快件申请单，申请号由系统自动生成。快件申请信息包括申请号、客户号、发件人、发件人电话、快件名称、运费、发出地、收件人、收件人电话、收件地址。其中，一个申请号对应唯一的一个快件申请，一个客户可以提交多个快件申请，但一个快件申请由唯一的一个客户提交。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图(图 7-4)和关系模式(不完整)如下：

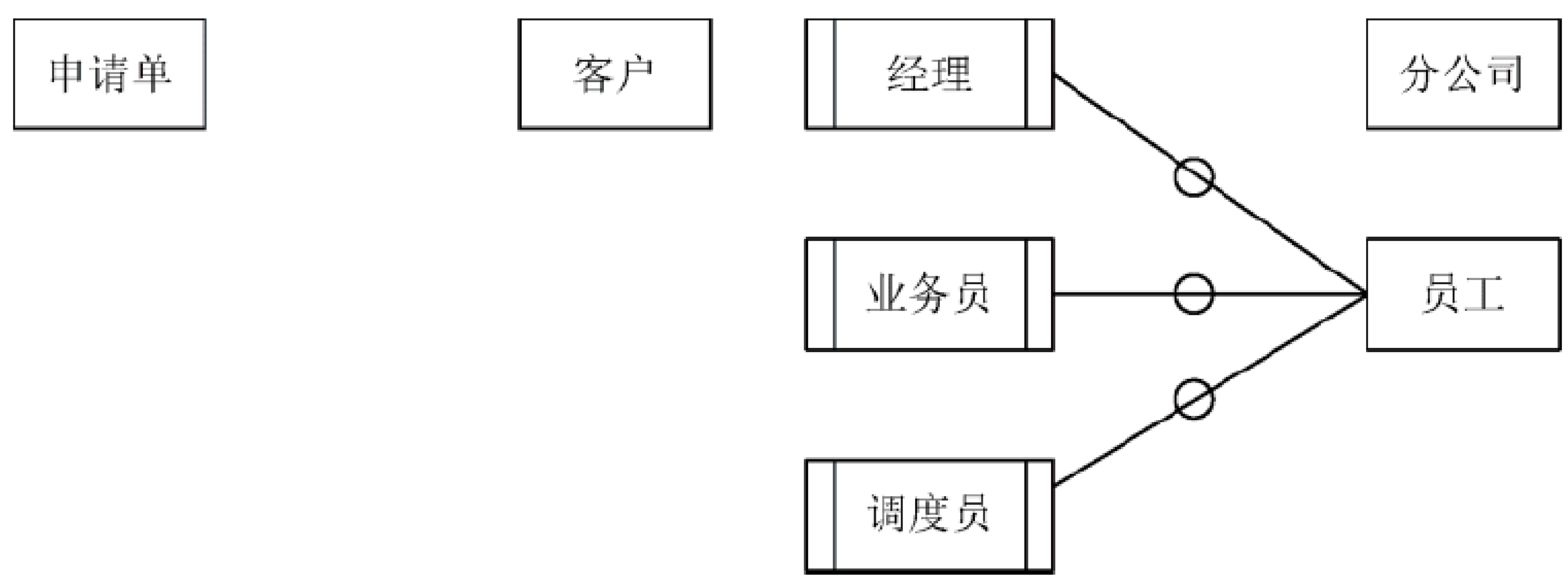


图 7-4　某物品运送信息管理系统实体联系图(不完整)

【关系模式设计】

分公司(分公司编号，名称，经理，办公电话，地址)

员工(员工号，姓名，___(a)___，岗位，薪资，手机号，家庭地址)

客户(客户号，单位名称，通信地址，所属省份，联系人，联系电话，银行账号)

申请单(___(b)___，发件人，发件人电话，发件人地址，快件名称，运费，收件人，收件人电话，收件地址，受理标志，业务员)

安排承运(___(c)___，实际完成时间，调度员)

【问题 1】

根据问题描述，补充 5 个联系，完善图 7-4 的实体联系图。联系名可用联系 1、联系 2、联系 3、联系 4 和联系 5 代替，联系的类型分为 1∶1、1∶n 和 m∶n(或 1∶1、1∶*和*∶*)。

【问题 2】

(1) 根据实体联系图，将关系模式中的空(a)～(c)补充完整；

(2) 给出员工、申请单和安排承运关系模式的主键和外键。

【问题 3】

(1) 客户关系的通信地址可以进一步分为邮编、省、市、街道，那么该属性是否属于简单属性，为什么？请用 100 字以内的文字说明。

(2) 假设分公司需要增设一位经理的职位，那么分公司与经理之间的联系类型应修改为__(d)__，分公司的主键应修改为__(e)__。

【解析】

本题考查数据库设计，设计考点有：数据库的概念结构设计和逻辑结构设计。

【问题 1】由“每个分公司有一位经理”可知分公司与经理之间的管理联系类型为 1:1；由“每个分公司有多名员工处理日常事务，每个员工属于一个分公司”可知分公司与员工间的所属联系类型为 1:*；并且员工是经理的超类型，经理是员工的子类型。

由“一个客户可以有多个快件申请，但一个快件申请对应唯一的一个客户”可知，客户与申请单之间的提交联系类型为 1:*。

由“业务员根据客户提交的快件申请单进行快件受理事宜，一个业务员可以受理多个客户的快件申请，一个快件申请只能由一个业务员受理”可知业务员与申请单之间的受理联系类型为 1:*。

由“调度根据已受理的申请单安排快件的承运事宜，例如：执行承运的业务员、运达时间等；一个业务员可以执行调度安排的多个快件的承运业务。”可知，调度、业务员和申请单之间的承运联系类型为 1:*:*。

【问题 2】逻辑结构设计中，分公司信息包括分公司编号、名称、经理、办公电话和地址。每个分公司可以有多名员工处理分公司的日常业务，故主键为员工号，外键为分公司编号。

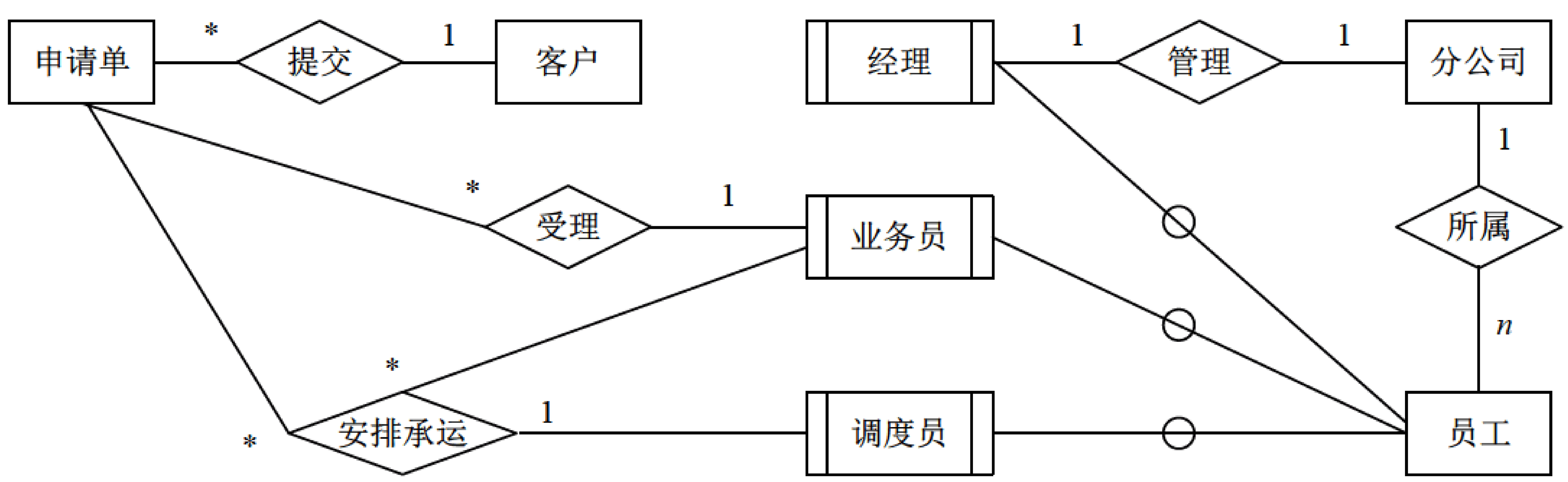
业务员根据客户提交的快件申请单进行快件受理事宜，一个业务员可以受理多个客户的快件申请，一个快件申请只能由一个业务员受理，故主键为申请号，外键为客户号、业务员；调度员根据已受理的申请单安排快件的承运事宜，例如：执行承运的业务员、运达时间等。一个业务员可以执行调度员安排的多个快件的承运业务。故主键为申请号，外键为业务员、调度员。

【问题 3】(1)简单属性是原子的、不可再分的，复合属性可以细分为更小的部分。根据题意，客户关系的通信地址可以进一步分为邮编、省、市、街道，该属性属于复杂属性。

(2)分公司需要增设一位经理的职位，分公司可以有多位经理，所以分公司与经理之间的联系类型应该修改为 1:n，分公司主键应修改为分公司编号、经理。

【参考答案】

【问题 1】



【问题 2】

(1) (a) 分公司编号

(b) 申请号，客户号

(c) 申请号，业务号

(2)

关系模式	主键	外键
员工	员工号	分公司编号
申请单	申请号	客户号，业务员
安排乘运	申请号	业务员，调度员

【问题 3】

(1) 该属性不属于简单属性。因为简单属性是原子的、不可再分的，而复合属性可以细分为更小的部分。客户关系的通信地址可以进一步分为邮编、省、市、街道，该属性属于复合属性。

(2) (d) 1:n;

(e) 分公司编号，经理

【典型题 7-12】(2013 年 5 月下午试题二)

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某电视台拟开发一套信息管理系统，以方便对全台的员工、栏目、广告和演播厅等进行管理。

【需求分析】

(1) 系统需要维护全台员工的详细信息、栏目信息、广告信息和演播厅信息等。员工的信息主要包括：工号、姓名、性别、出生日期、电话、住址等。栏目信息主要包括：栏目名称、播出时间、时长等。广告信息主要包括：广告编号、价格等。演播厅信息包括：房间号、房间面积等。

(2) 电视台分局调度单来协调各档栏目、演播厅和场务。一销售档栏目只会占用一个演播厅，但会使用多名场务来进行演出协调。演播厅和场务可以被多个栏目循环使用。

(3) 电视台根据栏目来插播广告。每档栏目可以插播多条广告，每条广告也可以在多档栏目插播。

(4) 一档栏目可以有多个主持人，但一名主持人只能主持一档栏目。

(5) 一名编辑人员可以编辑多条广告，一条广告只能由一名编辑人员编辑。

【概念模型设计】

根据需求阶段收集的信息设计的实体联系图(不完整)如图 7-5 所示。

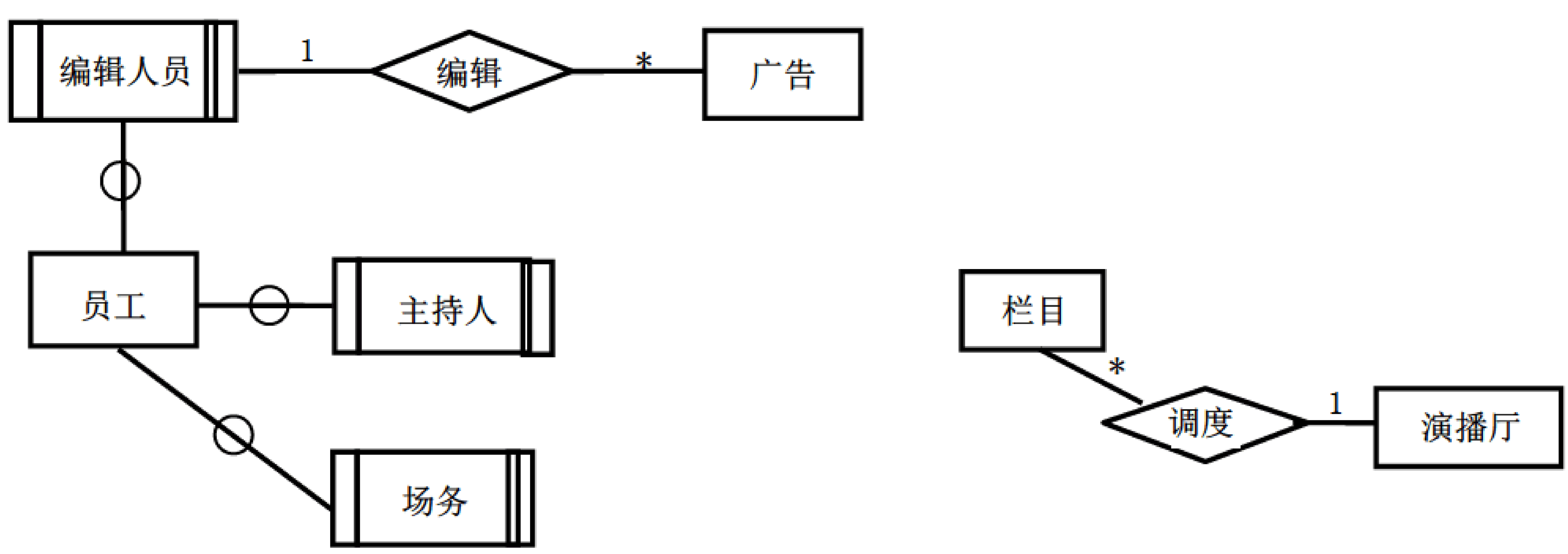


图 7-5　实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式(不完整)：

演播厅 (<u>房间号</u>，房间面积)
栏目 (<u>栏目名称</u>，播出时间，时长)
广告 (广告编号，销售价格，____(1)____)
员工 (<u>工号</u>，姓名，性别，出生日期，电话，住址)
主持人 (主持人工号，____(2)____)
插播单 (____(3)____，播出时间)
调度单 (____(4)____)

【问题 1】

补充图 7-4 中的联系和联系的类型。

【问题 2】

根据图 7-5，将逻辑结构设计阶段生成的关系模式的空(1)～(4)补充完整，并用下划线指出(1)～(4)所在关系模式的主键。

【问题 3】

现需要记录广告商信息，增加广告商实体。一个广告商可以提供多条广告，一条广告只由一个广告商提供。请根据该要求，对 7-5 进行修改，画出修改后的实体间联系和联系的类型。

【解析】

本题考查数据库设计，设计考点有：数据库的概念结构设计和逻辑结构设计。

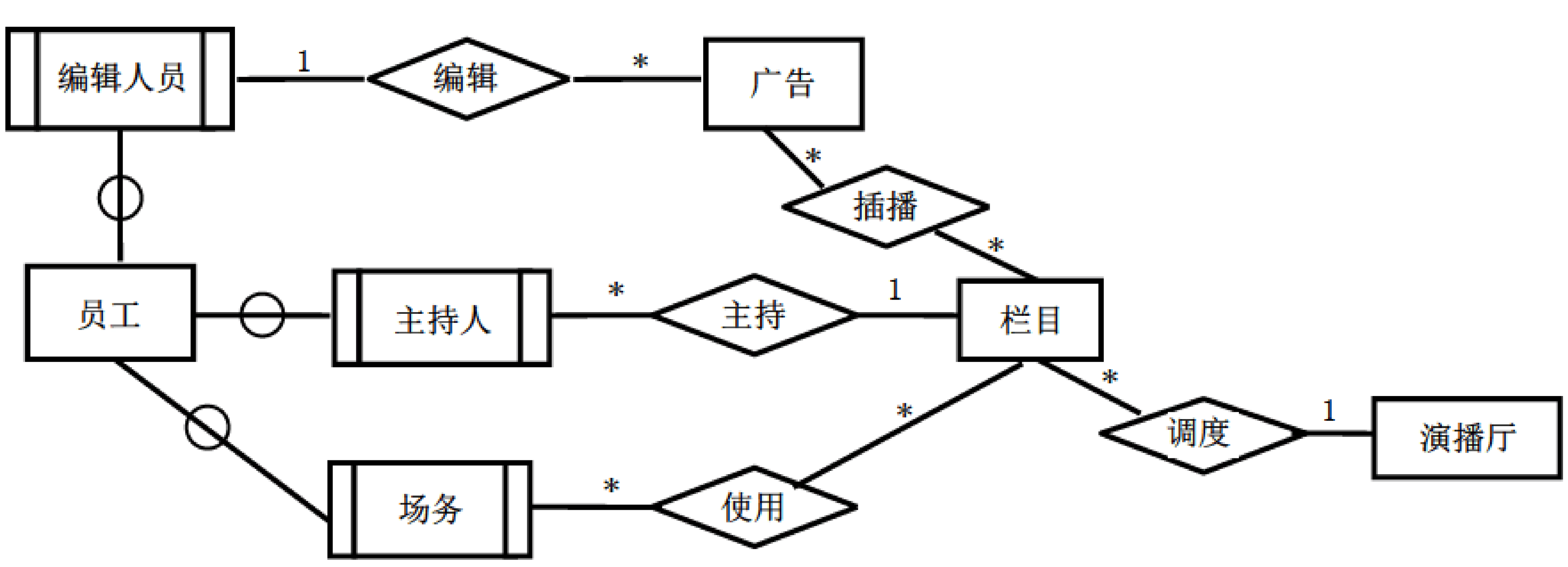
【问题 1】由说明每档栏目可以插播多条广告，每条广告可以在多档栏目中插播，可知广告和栏目之间是插播关系且为多比多；一个主持人可以主持一个栏目，一个栏目可以被多个主持人所主持，故主持人和栏目之间为多比一的关系；多名场务来进行栏目的演出协调，而场务又可以被多个栏目循环使用，故其关系为多比多。

【问题 2】逻辑结构设计中，广告实体中缺少广告时长，主键为广告编号；主持人实体与栏目实体为多比一的关系，故将栏目中主键栏目名称加入到主持人实体中，主键为主持人工号；插播单为栏目实体和广告实体这种多对多的关系所派生出的实体，其中记录了栏目和广告的主键信息，故插播单中缺少栏目名称和广告编号信息，主键为栏目名称和广告编号；调度单为场务、栏目和演播厅实体这种多对对的关系所派生的实体，故其记录了栏目名称，房间号，场务工号，主键为栏目名称、房间号和场务工号。

【问题 3】因为一个广告商可以提供多条广告，一条广告只能由一个广告商提供，故广告商和广告之间的关系为一比多。

【答案】

【问题 1】



【问题 2】

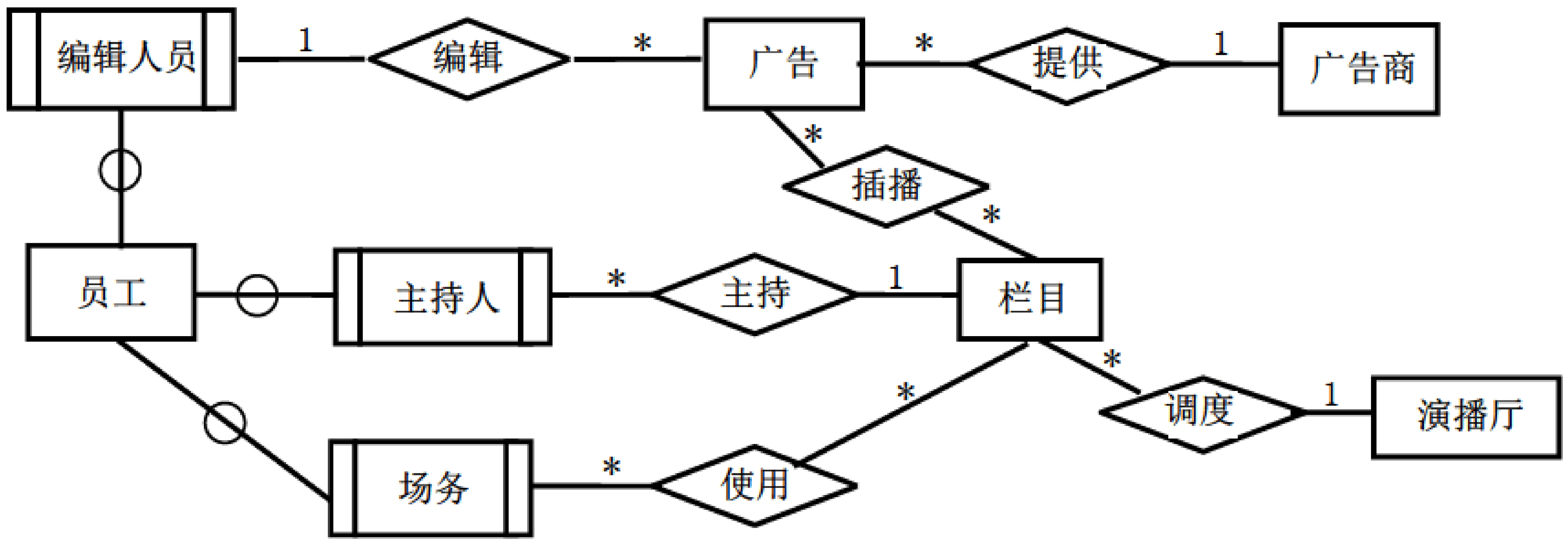
(1) 广告时长　　　　主键：广告编号

(2) 栏目名称　　　　　　　　主键：主持人工号

(3) 栏目名称、广告编号　　　主键：栏目名称、广告编号

(4) 栏目名称、房间号、场务工号　主键：栏目名称、房间号、场务工号

【问题 3】



【典型题 7-13】(2012 年 11 月下午试题二)

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某会议策划公司为了方便客户，便于开展和管理各项业务活动，需要构建一个基于网络的会议预定系统。

【需求分析】

(1) 会议策划公司设有受理部、策划部和其他部门，部门信息包括部门号、部门名称、部门主管、电话和邮箱号，每个部门有多名员工处理部门的日常事务，每名员工只能在一个部门工作，每个部门有一名主管负责管理本部门的事务和人员。

(2) 员工信息包括员工号、姓名、部门号、职位、联系方式和工资；其中，职位包括主管、业务员、策划员等。业务员负责受理会议申请，若申请符合公司规定，则置受理标志并填写业务员的员工号，策划部主管为已受理的会议申请制定策划任务，包括策划内容、参与人数、要求完成时间等。一个已受理的会议申请对应一个策划任务，一个策划任务只对应一个已受理的会议申请，但一个策划任务可由多名策划员参与执行，且一名策划员可以参与多项策划任务。

(3) 客户信息包括客户号、单位名称、通信地址、所属省份、联系人、联系电话、银行账号。其中，一个客户号唯一标识一个客户。一个客户可以提交多个会议申请，但一个会议申请对应唯一的一个客户号。

(4) 会议申请信息包括申请号、开会日期、会议地点、持续天数、会议人数、预算费用、会议类型、酒店要求、会议室要求、客房类型、客房数、联系人、联系方式、受理标志和业务员的员工号等。客房类型有豪华套房、普通套房、标准间、三人间等，且申请号和客房类型决定客房数。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式(不完整)如图 7-6 所示。

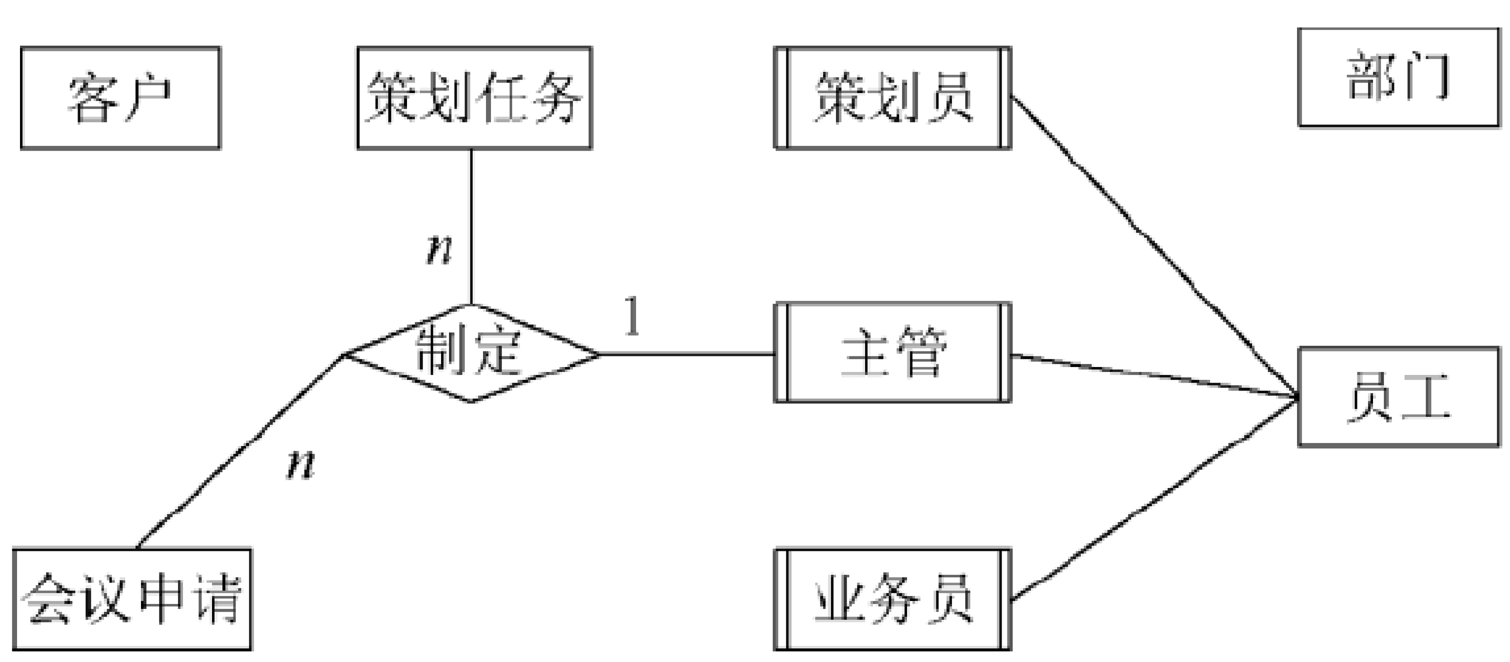


图 7-6　实体联系图

【关系模式设计】

部门(部门号，部门名称，主管，电话，邮箱号)

员工(员工号，姓名，__(a)__，联系方式，工资)

客户(客户号，单位名称，通信地址，所属省份，联系人，联系电话，银行账号)

会议申请(__(b)__，开会日期，会议地点，持续天数，会议人数，预算费用，会议类型，酒店要求，会议室要求，客房数，联系人，联系方式，受理标志，员工号)

策划任务(__(c)__，策划内容，参与人数，要求完成时间)

执行策划(__(d)__，实际完成时间)

【问题 1】

根据问题描述，补充五个联系、联系的类型，完善图 7-6 的实体联系图。

【问题 2】

根据实体联系图，将关系模式中的空(a)～(d)补充完整(1 个空缺处可能有多个数据项)。

对会议申请、策划任务和执行策划关系模式，用下划线和#分别指出各关系模式的主键和外键。

【问题 3】

请说明关系模式“会议申请”存在的问题及解决方案。

【解析】

本题考查数据库系统中实体联系模型和关系模式设计方面的应用知识，属于比较传统的题目，考查点也和往年类似。

【问题 1】

本题考查数据库的概念结构设计。两个实体集之间的联系类型分为三类：一对一(1∶1)联系、一对多(1∶n)联系和多对多(m∶n)联系。

根据题意，每个部门有多名员工，每个员工只能在一个部门工作，因此部门和员工之间存在一个名为“所属”的 1∶n 联系；每个部门只有一个主管，负责管理本部门的事务和人员，因此部门和主管之间存在一个名为“管理”的 m∶n 联系；一个策划任务可由多名策划员参与执行，而一名策划员可以参与多项策划任务，因此策划员和策划任务之间存在一个名为“执行”的 m∶n 联系；一个客户可以提交多个会议申请，一个会议申请对应唯一的客户，因此客户和会议申请之间存在一个名为“执行”的 1∶n 联系；一个业务员可以申请多个会议，一个会议对应一个业务员，因此业务员和会议申请之间存在一个名为“受理”的 1∶n 联系。

根据以上分析，完善图 7-6 所示的实体联系图。

【问题 2】

本问题考查数据库的逻辑结构设计，题目要求补充完整各关系模式，并给出各关系模式的主键和外键。

根据实体联系图和需求描述，员工关系模式主要包含员工号、姓名、部门号、职位、联系方式和工资等信息，因此(a)处应填入“部门号，职位”；会议申请主要包含申请号、开会日期、会议地点、持续天数、会议人数、预算费用、会议类型、酒店要求、会议室要求、客房类型、客房数、联系人、联系方式、受理标志、员工号等信息，又由于客户和会议申请间存在一对多联系，应将一端的键并入多端，因此(b)处应填入“申请号，客房类型，客户号”，其中主键为申请号，外键为客户号和员工号；由图 7-6 中策划任务与会议申请及主管之间的关系，应将另两个关系的键并入该关系，因此(c)处应填入“申请号，员工号”，其

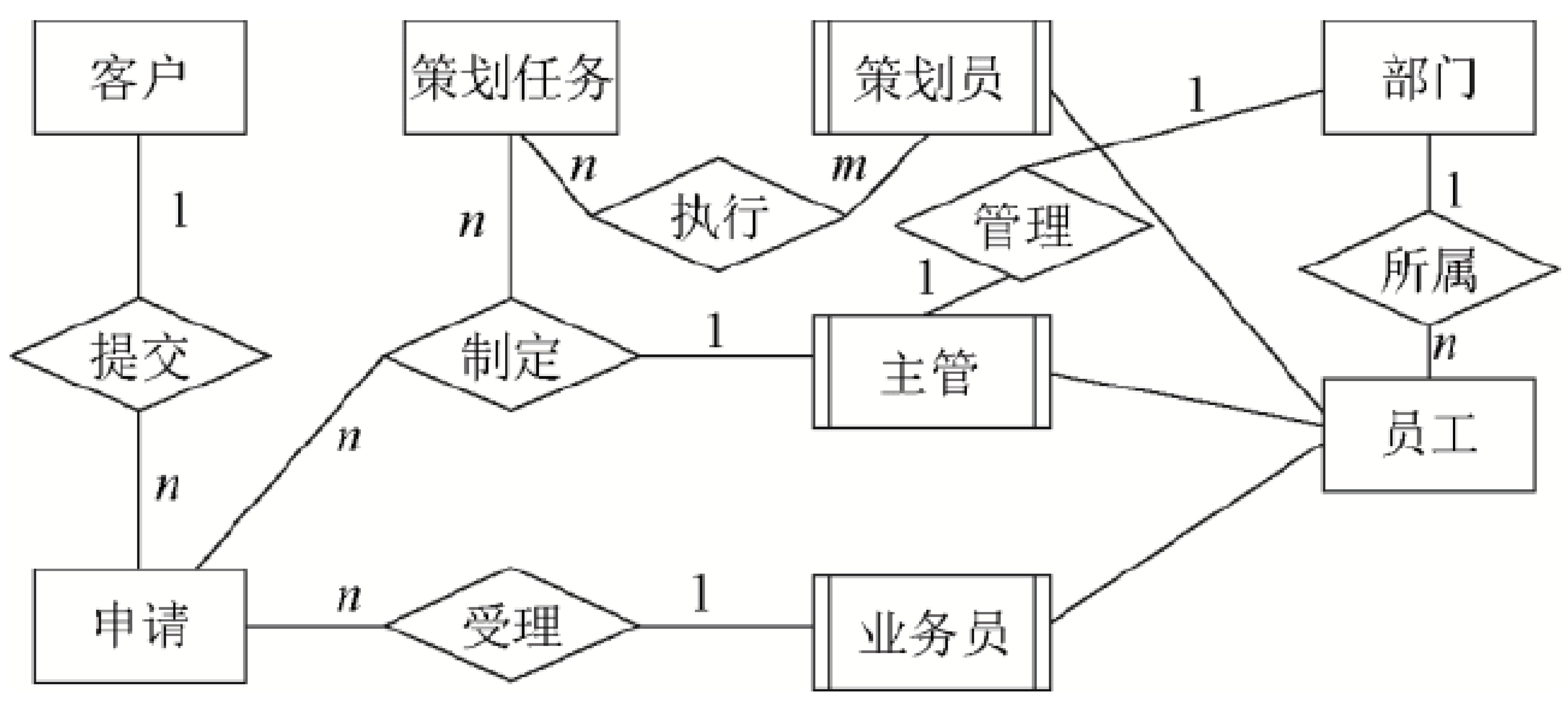
主键和外键为申请号和员工号；由策划任务、策划员和执行策划三者间的关系可知，(d)处应填入“申请号，员工号”，其主键为申请号、员工号和实际完成时间，外键为申请号和员工号。

【问题 3】

会议申请关系中的每个属性都是不可分割的数据项，且申请号和客房类型决定类型，“客房类型、客房数”等信息在会议申请关系中重复存储，存在大量的数据冗余问题。此外，若一部分信息被修改，而另一部分没有修改，容易造成数据修改的不一致性问题。

【参考答案】

【问题 1】



【问题 2】

(a)部门号，职位

(b)申请号，客房类型，客户号，其中主键为<u>申请号</u>，外键为客户号#、员工号#

(c)申请号，员工号，其中主键和外键均为申请号和员工号

(d)申请号，员工号，其中主键为<u>申请号，员工号，实际完成时间</u>，外键为申请号#，员工号#

【问题 3】

“会议申请”关系模式存在数据冗余及数据修改的不一致性问题，应将其分解为如下两个关系模式：会议申请 1(申请号，客户号，开会日期，会议地点，持续天数，会议人数，预算费用，会议类型，酒店要求，会议室要求，联系人，联系方式，受理标志，员工号)及会议申请 2(申请号，客房类型、客房数)。

【典型题 7-14】(2012 年 5 月下午试题二)

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某医院拟开发一套住院病人信息管理系统，以方便对住院病人、医生、护士和手术等信息进行管理。

【需求分析】

(1) 系统登记每个病人的住院信息，包括：病案号，病人的姓名、性别、地址、身份证号、电话号码、入院时间及病床号信息，每个病床有唯一所属的病区及病房，如表 7-8 所示。其中病案号唯一标识病人本次住院的信息。

表 7-8　住院登记表

病案号	071002286	姓名	张三	性别	男
身份证号	0102196701011234	入院时间	2011-03-03	病床号	052401
病房	0524室	病房类型	三人间	所属病区	05 II区

(2) 在一个病人的一次住院期间，由一名医生对该病人的病情进行诊断，并填写一份诊断书，如表 7-9 所示。对于需要进行一次或多次手术的病人，系统记录手术名称、手术室、手术日期、手术时间、主刀医

生及多名协助医生，每名医生在手术中的责任不同，如表 7-10 所示，其中手术室包含手术室号、楼层、地点和类型等信息。

表 7-9　诊断书

诊断时间：2011年3月

病案号	071002286	姓名	张三	性别	男	医生	李**
诊断							

表 7-10　手术安排表

手术名称	**手术	病案号	071002286	姓名	张三	性别	男
手术室	032501	手术日期	2011-03-15	手术时间	8:30～10:30	主刀医生	李**
协助医生	王**(协助)，周**(协助)，刘**(协助)，高**(麻醉)						

(3) 护士分为两类：病床护士和手术室护士。每个病床护士负责护理一个病区内的所有病人，每个病区由多名护士负责护理。手术室护士负责手术室的护理工作。每个护士负责多个手术室，每个手术室由多名护士负责，每个护士在手术室中有不同的责任，并由系统记录其责任。

【概念模型设计】

根据需求阶段收集的信息，设计的实体联系图(不完整)如图 7-7 所示。

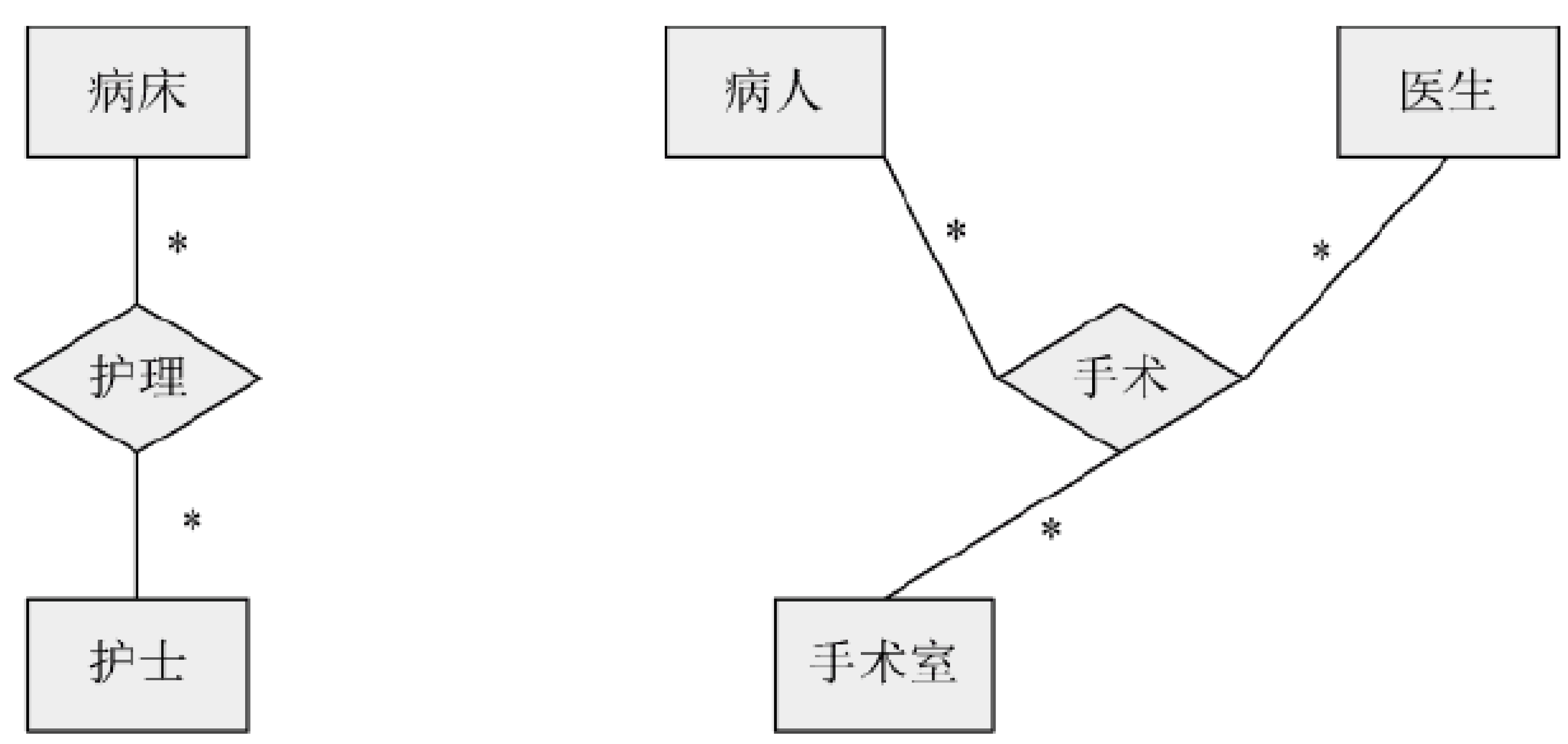


图 7-7　实体联系图

【逻辑结构设计】

根据概念模型设计阶段完成的实体联系图，得出如下关系模式(不完整)：

病床(病床号，病房，病房类型，所属病区)

护士(护士编号，姓名，类型，性别，级别)

病床护士((1))

手术室(手术室号，楼层，地点，类型)

手术室护士((2))

病人((3) ，姓名，性别，地址，身份证号，电话号码，入院时间)

医生(医生编号，姓名，性别，职称，所属科室)

诊断书((4) ，诊断，诊断时间)

手术安排(病案号，手术室号，手术时间，手术名称)

手术医生安排((5) ，医生责任)

【问题 1】

补充图 7-7 中的联系和联系的类型。

【问题 2】

根据图 7-7，将逻辑结构设计阶段生成的关系模式中的空(1)～(5)补充完整，并用下划线指出主键。

【问题 3】

如果系统还需要记录医生给病人的用药情况，即记录医生给病人所开处方中药品的名称、用量、价格、药品的生产厂家等信息。请根据该要求，对图 7-7 进行修改，画出补充后的实体、实体间联系和联系的类型。

【解析】

【问题 1】病人住院才会安排病床，且一个病人只安排在某一个病床上，不停地有病人住院、出院，某一病床上的病人也在变，由此可见，病床和病人之间的联系是一对多。

病人住院期间，由一名医生对其病情进行诊断，一名医生可以诊断多名病人，医生和病人之间的联系是一对多。

每个手术室护士负责多个手术室，每个手术室由多名护士负责，护士和手术室之间的联系是多对多。

【问题 2】病床护士负责护理一个病区内的所有病人，每个病区由多名护士负责护理。因此，病床护士的属性有：护士编号、病区。病区和病床护士之间的联系是一对多，由病床护士编号可以推出其所属病区，因此关键字为护士编号。

每个护士负责多个手术室，每个手术室由多名护士负责，每个护士在手术室中有不同的责任，并由系统记录其责任。因此手术室护士的属性有：护士编号、手术室号、责任。手术室和手术室护士之间是多对多的联系，每个护士的责任由其编号和所在的手术室号一起决定，因此护士编号和手术室号一起构成关键字。

病人入院时需要登记病案号、病人的姓名、性别、地址、身份证号、电话号码、入院时间及病床号。病人实体的属性少病案号和病床号。而病案号唯一标识病人本次住院的信息，因此病案号是关键字。

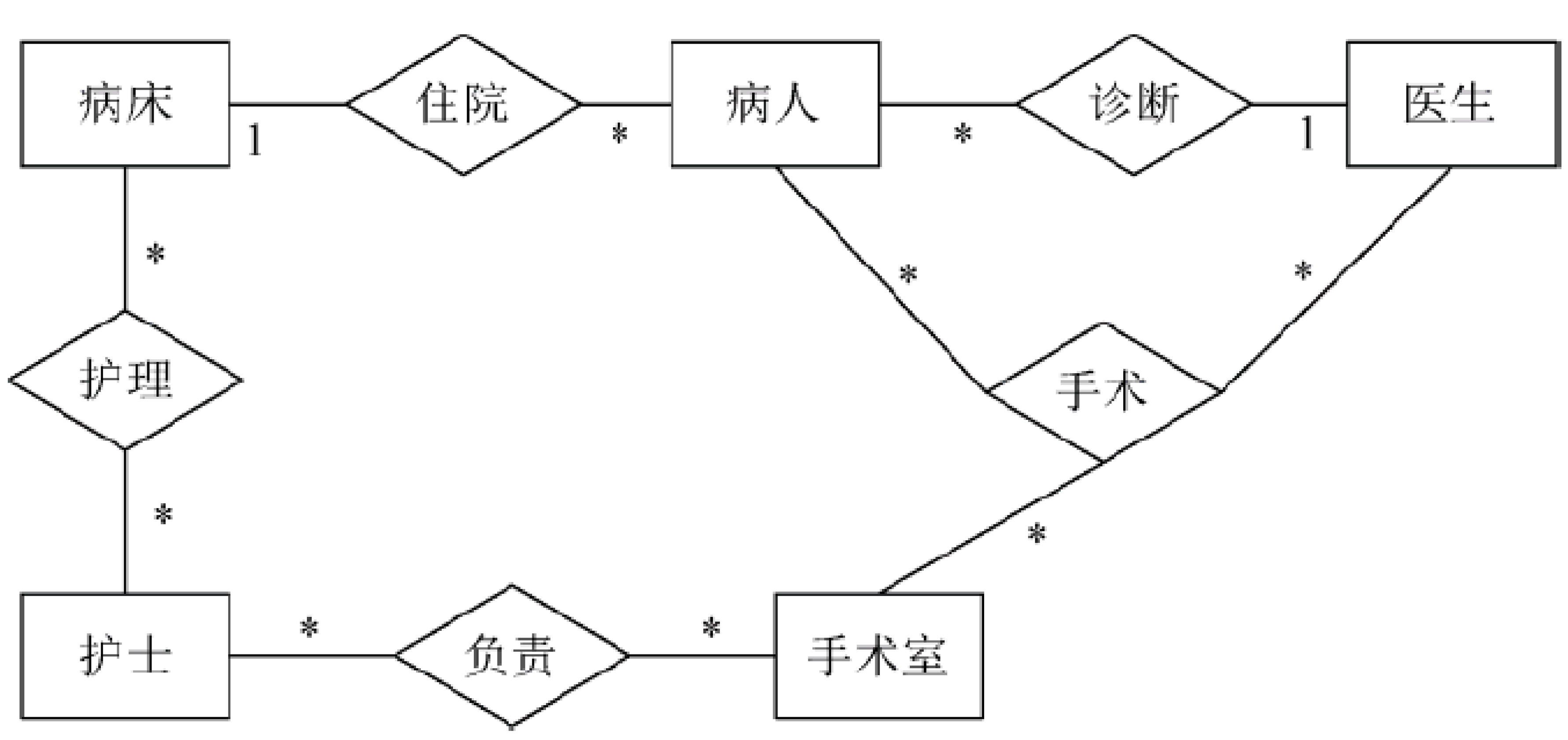
诊断书是医生对病人病情的诊断，医生由医生编号唯一标识，病人由病案号唯一标识，诊断书实体应为诊断书(病案号，医生编号，诊断，诊断时间)。一份诊断书只属于一个病人，因此诊断书可以由病案号唯一标识，病案号为关键字。

手术医生安排是要根据病人来进行的，而一名病人可能要进行 1 次或多次手术，每次手术的时间不同，所在手术室可能不同。手术医生安排实体为：手术医生安排(病案号，手术室号，手术时间，医生编号，医生责任)。

【问题 3】病人在不同的阶段，可能会有不同的医生开处方。一名医生不同阶段所开的处方也会不同，因此就处方关系而言，医生和病人之间是多对多的联系。

【答案】

【问题 1】



【问题2】(1)病区，<u>护士编号</u>

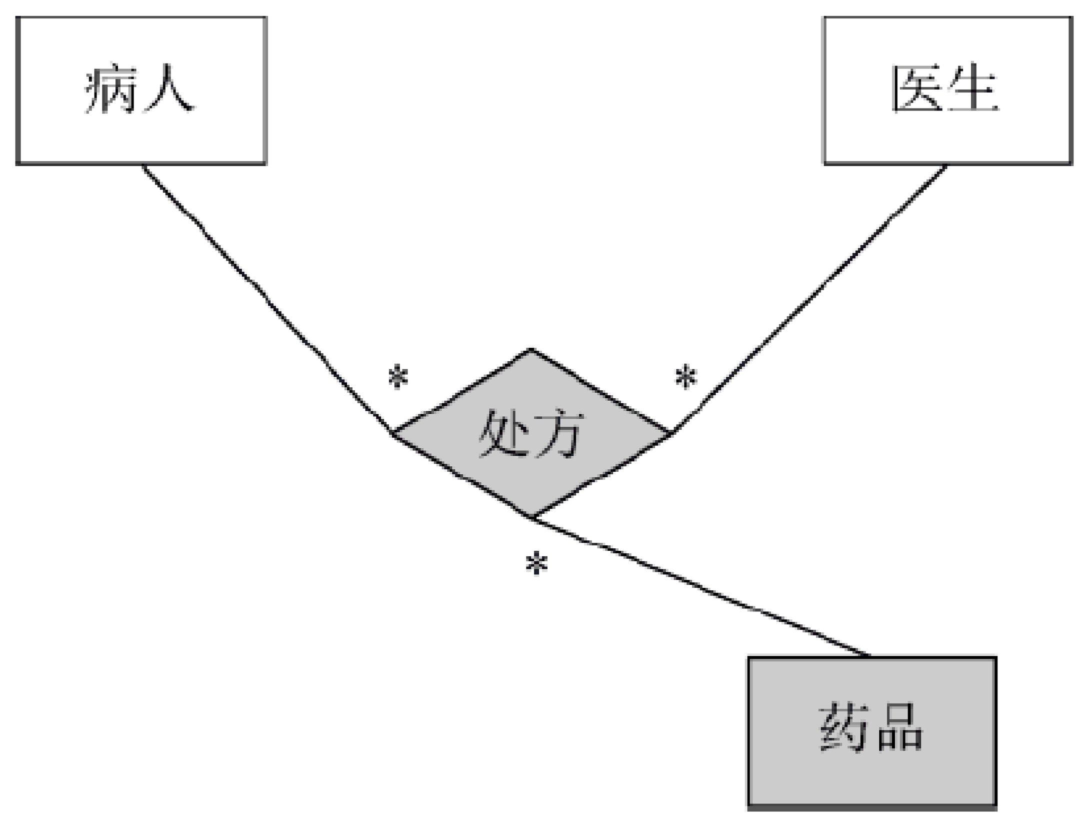
(2)<u>手术室号，护士编号</u>，责任

(3)<u>病案号</u>，病床号

(4)<u>病案号</u>，医生编号

(5)<u>病案号，手术室号，手术时间，医生编号</u>

【问题3】



7.9 过 关 习 题

1. 设有学生实体 Students(学号，姓名，性别，年龄，家庭住址，家庭成员，关系，联系电话)，其中“家庭住址”记录了邮编、省、市、街道信息；“家庭成员，关系，联系电话”分别记录了学生亲属的姓名、与学生的关系以及联系电话。

学生实体 Students 中的“家庭住址”是一个_(1)_属性；为使数据库模式设计更合理，对于关系模式 Students_(2)_。

(1) A. 简单　　B. 多值　　C. 复合　　D. 派生

(2) A. 可以不作任何处理，因为该关系模式达到了 3NF

B. 只允许记录一个亲属的姓名、与学生的关系以及联系电话的信息

C. 需要对关系模式 Students 增加若干组家庭成员、关系及联系电话字段

D. 应该将家庭成员、关系及联系电话加上学生号，设计成为一个独立的实体

2. 医院数据库的部分关系模式为：科室(科室号，科室名，负责人，电话)、病患(病历号，姓名，住址，联系电话)和职工(职工号，职工姓名，科室号，住址，联系电话)。假设每个科室有一位负责人和一部电话，每个科室有若干名职工，一名职工只属于一个科室；一个医生可以为多个病患看病；一个病患可以由多个医生多次诊治。科室与职工的所属联系类型是_(1)_，病患与医生的就诊联系类型为_(2)_。对于就诊联系最合理的设计是_(3)_，就诊关系的主键是_(4)_。

(1) A. 1∶1　　B. 1∶n　　C. n∶1　　D. n∶m

(2) A. 1∶1　　B. 1∶n　　C. n∶1　　D. n∶m

(3) A. 就诊(病历号，职工号，就诊情况)

B. 就诊(病历号，职工姓名，就诊情况)

C. 就诊(病历号，职工号，就诊时间，就诊情况)

D. 就诊(病历号，职工姓名，就诊时间，就诊情况)

(4) A. 病历号，职工号　　B. 病历号，职工号，就诊时间

C. 病历号，职工姓名　　　　　　　　D. 病历号，职工姓名，就诊时间

3. 若有关系 R(A, B, C, D)和 S(C, D, E)，则与表达式 $\Pi_{3,4,7}(\sigma_{4<5}(R\times S))$ 等价的 SQL 语句如下：

```
SELECT  (1)  FROM  (2)  WHERE  (3) ;
```

(1) A. A,B,C,D,E　　B. C,D,E
　C. R.A, R.B, R.C, R.D, S.E　　D. R.C, R.D, S.E

(2) A. R　　B. S　　C. R,S　　D. RS

(3) A. D<C　　B. R.D<S.C　　C. R.D<R.C　　D. S.D<R.C

4. 将 Students 表的插入权限赋予用户 UserA，并允许其将权限授予他人，应使用的 SQL 语句为：

```
GRANT  (1)  TABLE Students TO UserA  (2) ;
```

(1) A. UPDATE　　B. UPDATE ON　　C. INSERT　　D. INSERT ON

(2) A. FOR ALL　　B. PUBLIC
　C. WITH CHECK OPTION　　D. WITH GRANT OPTION

5. 给定关系模式 R<U, F>，U={A, B, C}，F={AB→C, C→B}。关系 R__(1)__，且分别有__(2)__。

(1) A. 只有 1 个候选关键字 AC　　B. 只有 1 个候选关键字 AB
　C. 有 2 个候选关键字 AC 和 BC　　D. 有 2 个候选关键字 AC 和 AB

(2) A. 1 个非主属性和 2 个主属性　　B. 2 个非主属性和 1 个主属性
　C. 0 个非主属性和 3 个主属性　　D. 3 个非主属性和 0 个主属性

6. 设有关系模式 R (课程，教师，学生，成绩，时间，教室)，其中函数依赖集 F 如下：

F= {课程→教师，(学生，课程)→成绩，(时间，教室)→课程，
　(时间，教师)→教室，(时间，学生)→教室}

关系模式 R 的一个主键是__(1)__，R 规范化程度最高达到__(2)__。若将关系模式 R 分解为 3 个关系模式 R1(课程，教师)、R2(学生，课程，成绩)、R3(学生，时间，教室，课程)，其中 R2 的规范化程度最高达到__(3)__。

(1) A. (学生，课程)　　B. (时间，教室)　　C. (时间，教师)　　D. (时间，学生)

(2) A. 1NF　　B. 2NF　　C. 3NF　　D. BCNF

(3) A. 2NF　　B. 3NF　　C. BCNF　　D. 4NF

7. E-R 图转换为关系模型时，对于实体 E1 与 E2 间的多对多联系，应该将______。

A. E1 的码加上联系上的属性并入 E2
B. E1 的码加上联系上的属性独立构成一个关系模式
C. E2 的码加上联系上的属性独立构成一个关系模式
D. E1 与 E2 的码加上联系上的属性独立构成一个关系模式

8. 在某企业的营销管理系统设计阶段，属性“员工”在考勤管理子系统中被称为“员工”，而在档案管理子系统中被称为“职工”，这类冲突称为______冲突。

A. 语义　　B. 结构　　C. 属性　　D. 命名

9. 阅读下列说明，回答问题 1 至问题 3。

【说明】

某物流公司为了整合上游供应商与下游客户，缩短物流过程，降低产品库存，需要构建一个信息系统

以方便管理其业务运作活动。

【需求分析】

(1) 物流公司包含若干部门，部门信息包括部门号、部门名称、经理、电话和邮箱。一个部门可以有多名员工处理部门的日常事务，每名员工只能在一个部门工作。每个部门有一名经理，只需负责本部门的事务和人员。

(2) 员工信息包括员工号、姓名、职位、电话号码和工资；其中，职位包括经理、业务员等。业务员根据托运申请负责安排承运货物事宜，例如装货时间、到达时间等。一个业务员可以安排多个托运申请，但一个托运申请只由一个业务员处理。

(3) 客户信息包括客户号、单位名称、通信地址、所属省份、联系人、联系电话、银行账号，其中，客户号唯一标识客户信息的每一个元组。每当客户要进行货物托运时，先要提出货物托运申请。托运申请包括申请号、客户号、货物名称、数量、运费、出发地、目的地。其中，一个申请号对应唯一的一个托运申请；一个客户可以有多个货物托运申请，但一个托运申请对应唯一的一个客户号。

【概念模式设计】

根据需求阶段收集的信息，设计的实体联系图和关系模式(不完整)如图 7-8 所示。

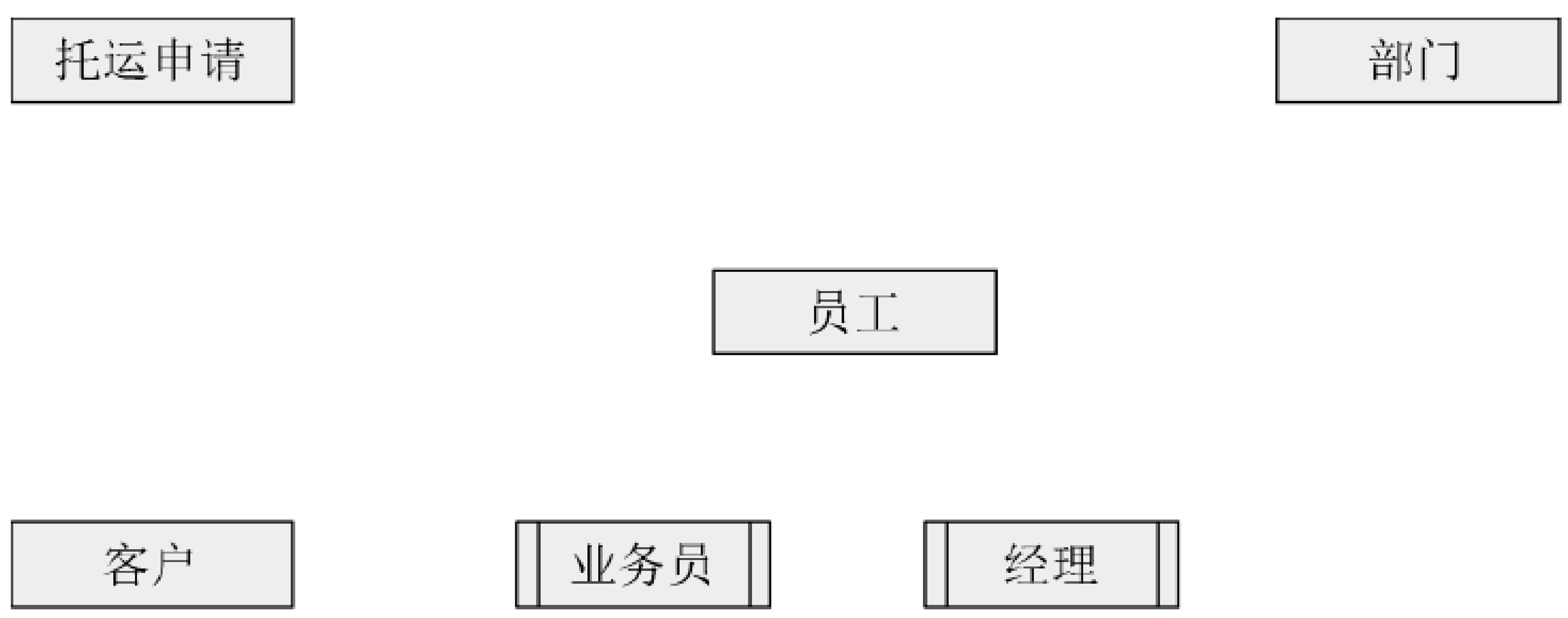


图 7-8　实体联系图

【关系模式设计】

部门(部门号，部门名称，经理，电话，邮箱)

员工(员工号，姓名，职位，电话号码，工资，__(a)__)

客户(__(b)__，单位名称，通信地址，所属省份，联系人，联系电话，银行账号)

托运请求(__(c)__，货物名称，数量，运费，出发地，目的地)

安排承运(__(d)__，装货时间，到达时间，业务员)

【问题 1】

根据问题描述，补充四个联系、联系的类型，以及实体与子实体的联系，完善图 7-8 所示的实体联系图。

【问题 2】

根据实体联系图，将关系模式中的空(a)～(d)补充完整。分别指出部门、员工和安排承运关系的主键和外键。

【问题 3】

若系统新增需求描述如下。

为了数据库信息的安全性，公司要求对数据库操作设置权限管理功能，当员工登录系统时，系统需要检查员工的权限。权限的设置人是部门经理。为了满足上述需求，应如何修改(或补充)图 7-8 所示的实体联系图，给出修改后的实体联系图和关系模式。

10. 阅读下列说明，回答问题 1 至问题 3。

【说明】

某服装销售公司拟开发一套服装采购管理系统，以便对服装采购和库存进行管理。

【需求分析】

(1) 采购系统需要维护服装信息及服装在仓库中的存放情况，服装信息主要包括服装编码、服装描述、服装类型、销售价格、尺码和面料，其中，服装类型为销售分类，服装按销售分类编码。仓库信息包括仓库编码、仓库位置、仓库容量和管理员。系统记录库管员的库管员编码、姓名和级别。一个库管员可以管理多个仓库，每个仓库有一名库管员。一个仓库中可以存放多类服装，一类服装可能存放在多个仓库中。

(2) 当库管员发现一类或者多类服装缺货时，需要生成采购订单。一个采购订单可以包含多类服装。每类服装可由多个不同的供应商供应，但具有相同的服装编码。采购订单主要记录订单编码、订货日期和应到货日期，并详细记录所采购的每类服装的数量、采购价格和对应的多个供应商。

(3) 系统需记录每类服装的各个供应商信息和供应情况。供应商信息包括供应商编码、供应商名称、地址、企业法人和联系电话。供应情况记录供应商所供应服装的服装类型和服装质量等级。一个供应商可以供应多类服装，一类服装可由多个供应商供应。库管员根据入库时的服装质量情况，设定或修改每个供应商所供应的每类服装的服装质量等级，作为后续采购服装时选择供应商的参考标准。

【概念模式设计】

根据需求阶段收集的信息，设计的实体联系图(不完整)如图 7-9 所示。

【逻辑结构设计】

根据概念设计阶段完成的实体联系图，得出如下关系模式(不完整)：

库管员(库管员编码，姓名，级别)

仓库信息((1) ，仓库位置，仓库容量)

服装(服装编码，服装描述，服装类型，尺码，面料，销售价格)

供应商(供应商编码，供应商名称，地址，联系电话，企业法人)

供应情况((2) ，服装质量等级)

采购订单((3))

采购订单明细((4))

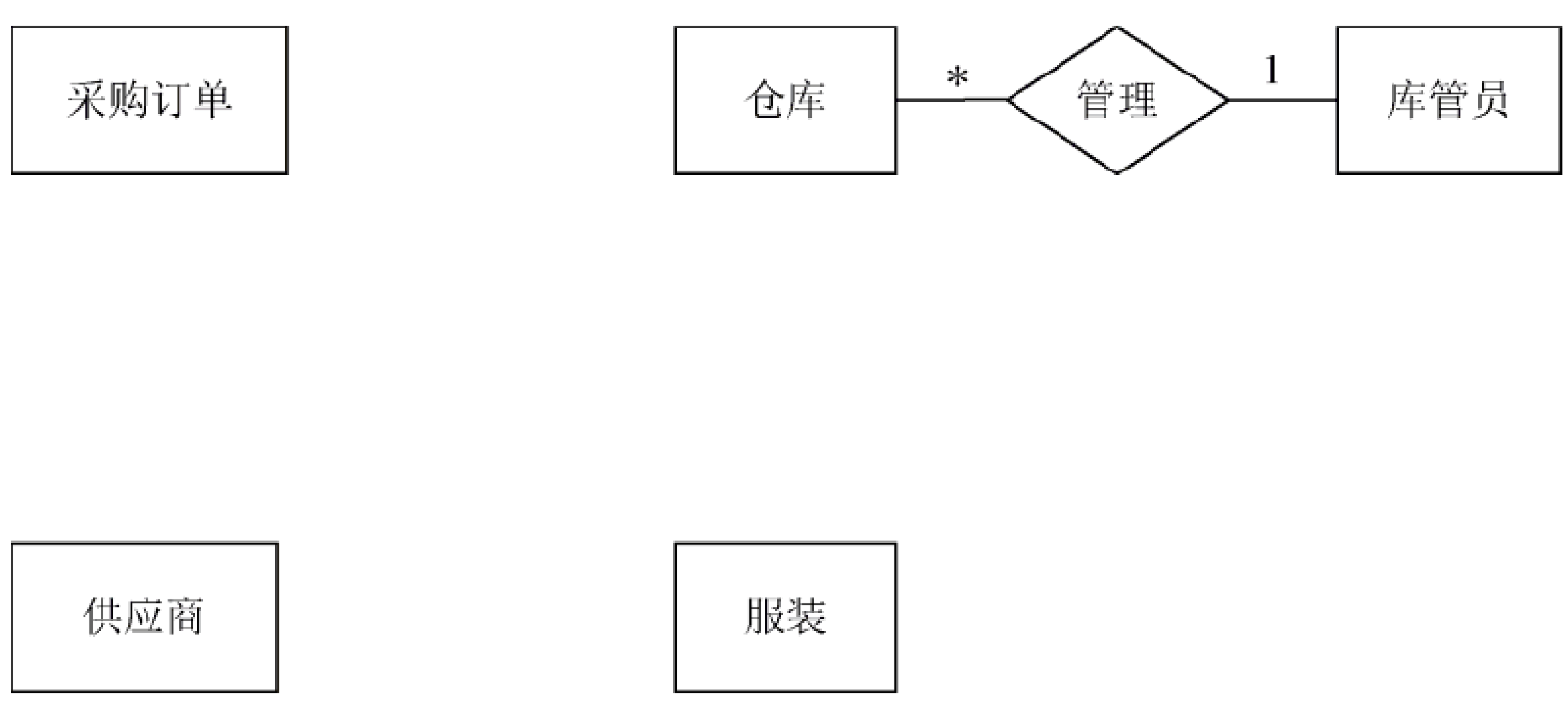


图 7-9　实体联系图

【问题 1】

根据需求分析的描述，补充图 7-9 中的联系和联系的类型。

【问题 2】

根据补充完整的图 7-9，将逻辑结构设计阶段生成的关系模式中的空(1)～(4)补充完整，并给出其主键(用下划线指出)。

【问题 3】

如果库管员需要定期轮流对所有仓库中的服装质量进行抽查，对每个仓库中的每一类被抽查服装需要记录一条抽查结果，并且需要记录抽查的时间和负责抽查的库管员。根据该要求，对图 7-9 进行修改，画出修改后的实体联系和联系的类型。

第8章

数据结构

本章要点

- 线性表
- 链表(单向链表、双向链表、循环链表)
- 栈和队列
- 串
- 数组、矩阵和广义表，数据压缩算法
- 树(二叉树、查找树、线索树、哈夫曼树)
- 图的定义、存储和操作，图的相关算法
- 各种排序算法
- 各种查找算法

8.1 线性结构

考核说明：本节主要介绍线性表、栈和队列的概念、存储结构、插入和删除运算等内容，同时介绍串的基本运算、存储结构和匹配模式。各种结构的存储方式和基本运算是考核重点，要能理解，并掌握相关的 C 语言算法描述。

8.1.1 线性表

1. 线性表的定义

线性表是 n 个元素的有限序列，通常记为$(a_1, a_2, \cdots, a_n)$。其特点如下。

- 存在唯一的一个称作“第一个”的元素。
- 存在唯一的一个称作“最后一个”的元素。
- 除了表头外，表中的每一个元素均只有唯一的直接前驱。
- 除了表尾外，表中的每一个元素均只有唯一的直接后继。

2. 线性表的存储结构

1) 顺序存储

线性表的顺序存储是用一组地址连续的存储单元依次存储线性表中的数据元素，从而使得逻辑关系相邻的两个元素在物理位置上也相邻。在这种存储方式下，存储逻辑关系无须占用额外的存储空间。其优点是可以随机存取表中的元素，缺点是插入和删除操作需要移动大量的元素。

一般地，在线性表的顺序存储结构中，第 i 个元素 a_i 的存储位置为

$$\mathrm{LOC}(a_i) = \mathrm{LOC}(a_1)+(i-1)\times L$$

其中，$\mathrm{LOC}(a_1)$是表中第一个元素的存储位置，L 是表中每个元素所占空间的大小。

2) 链式存储

线性表的链式存储是指用结点来存储数据元素，结点的空间可以是连续的，也可以是不连续的，因此存储数据元素的同时必须存储元素之间的逻辑关系。结点空间只有在需要的时候才申请，无须事先分配。最基本的结点结构如图 8-1 所示。

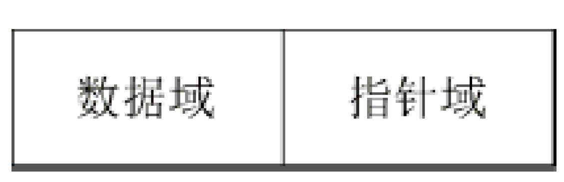


图 8-1 最基本的结点结构

其中，数据域用于存储数据元素的值，指针域则存储当前元素的直接前驱或直接后继信息，指针域中的信息称为指针(链)。n 个结点通过指针连成一个链表，若结点中只有一个指针域，则称为线性链表(单链表)。

线性表采用链表作为存储结构时，不能进行数据元素的随机访问，但其优点是插入和删除操作不需要移动元素。以下是几种其他链表结构。

(1) 双向链表：每个结点包含两个指针，指明直接前驱和直接后继元素，可在两个方

向上遍历链表。

(2) 循环链表：表尾结点的指针指向表中的第一个结点，可在任意位置开始遍历整个链表。

(3) 静态链表：借助数组来描述线性表的链式存储结构。

在链式存储结构中，只需要一个指针(头指针)指向第一个结点，就可以顺序访问表中的任意一个元素。为了简化对链表状态的判定和处理，特别引入一个不存储数据元素的结点，称为头结点，将其作为链表的第一个结点并令头指针指向该结点。

3. 线性表的插入和删除运算

1) 基于顺序存储结构的运算

插入元素前要移动元素以挪出空的存储单元，然后再插入元素；删除元素时同样需要移动元素，以填充被删除元素原来占有的存储单元。在等概率下平均移动元素的次数分别是

$$E_{\text{insert}}=\sum_{i=1}^{n+1}P_i\times(n-i+1)=\frac{1}{n+1}\sum_{i=1}^{n+1}(n-i+1)=\frac{n}{2}$$

$$E_{\text{delete}}=\sum_{i=1}^{n}q_i\times(n-i)=\frac{1}{n}\sum_{i=1}^{n}(n-i)=\frac{n-1}{2}$$

2) 基于链式存储结构的运算

在链式存储结构下进行插入和删除，其实质都是对相关指针的修改。

- 在单向链表中插入结点时，指针的变化情况如图 8-2 所示。
- 在单向链表中删除结点时，指针的变化情况如图 8-3 所示。

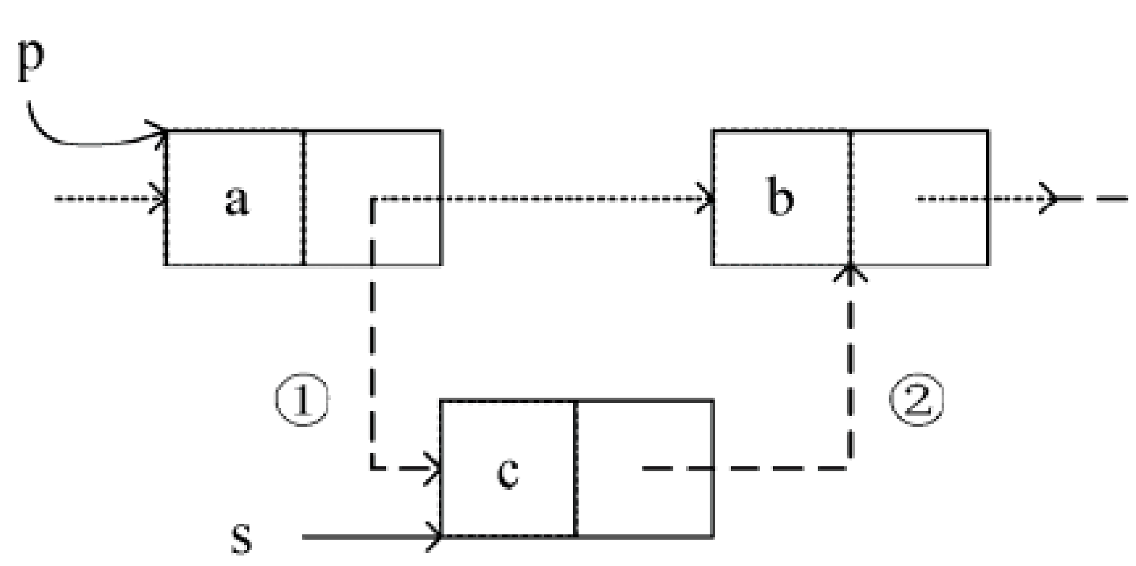


图 8-2　单向链表插入结点时的指针变化情况

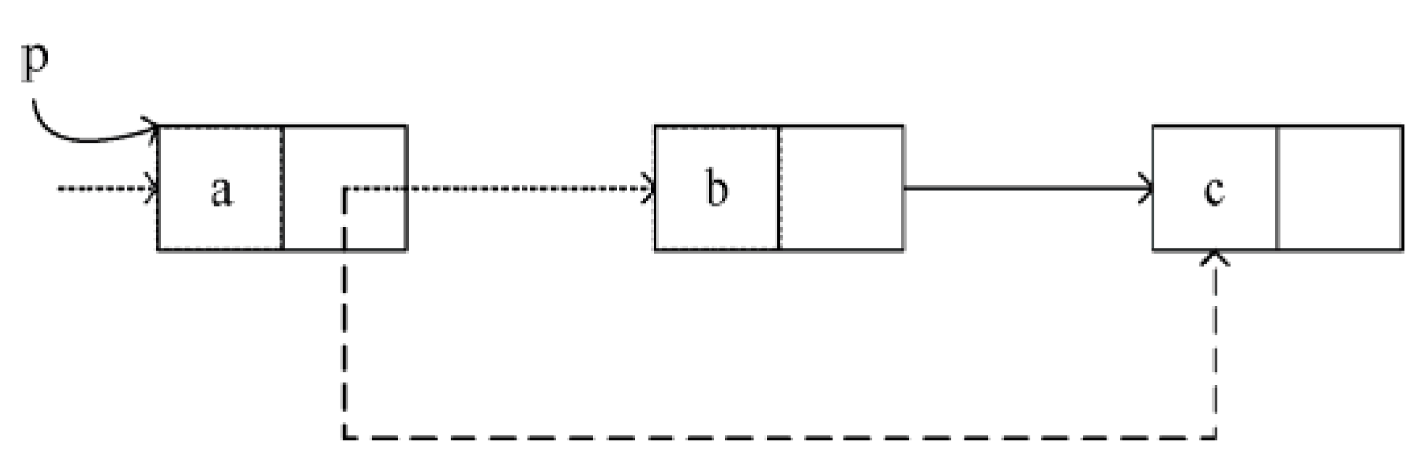


图 8-3　单向链表删除结点时的指针变化情况

- 在双向链表中插入结点时，指针的变化情况如图 8-4 所示。
- 在双向链表中删除结点时，指针的变化情况如图 8-5 所示。

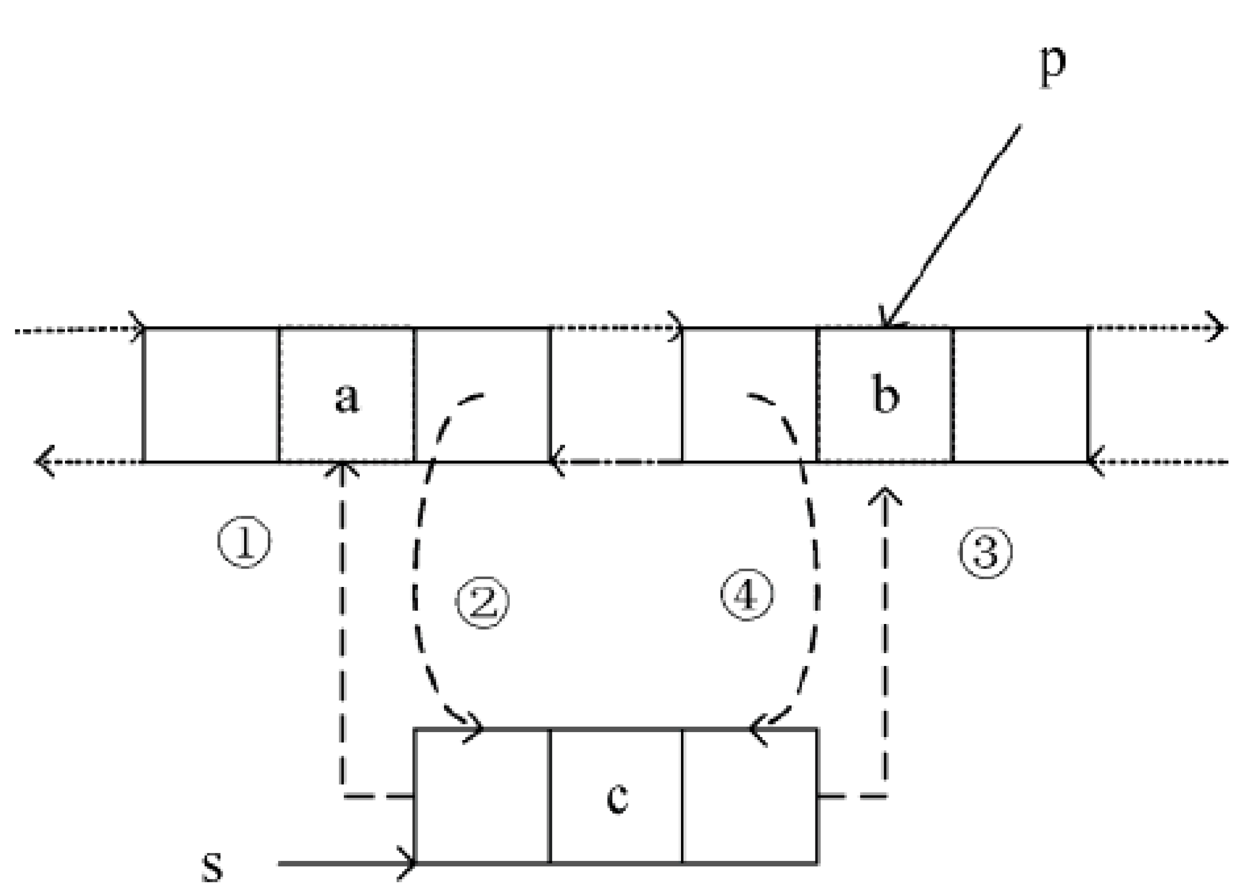


图 8-4　双向链表插入结点时的指针变化情况

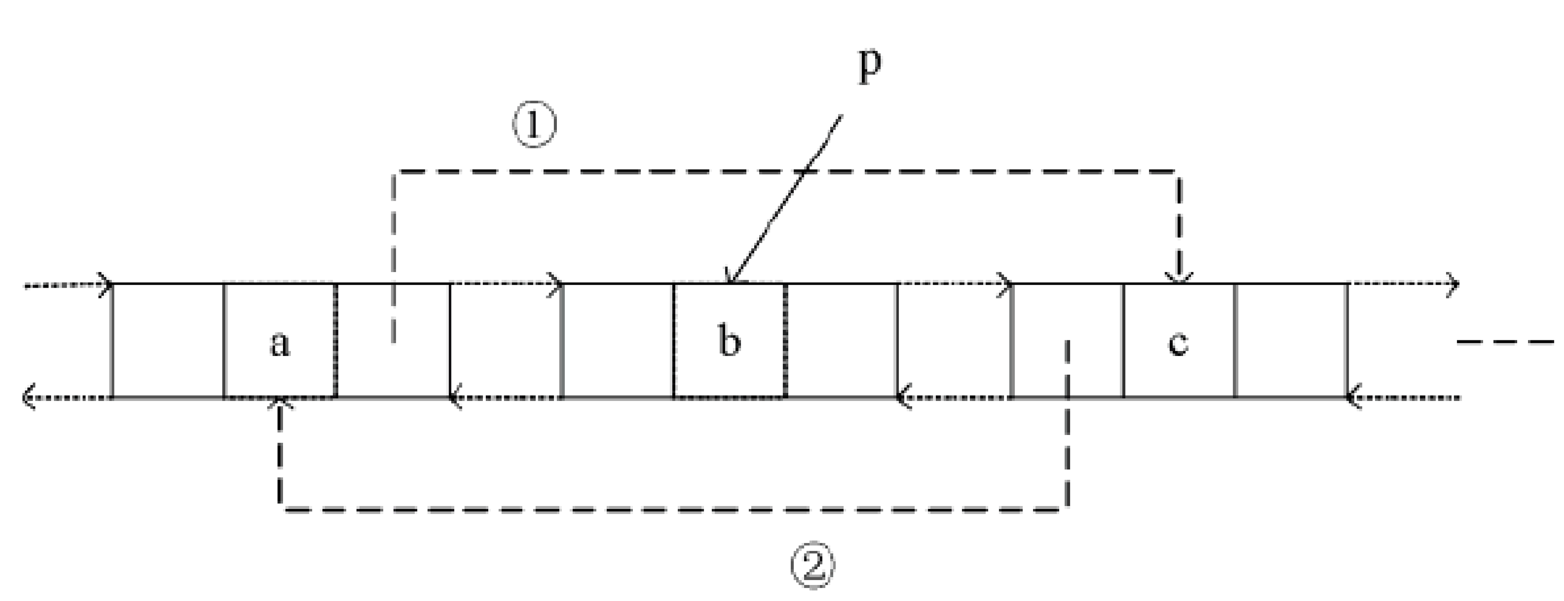


图 8-5　双向链表删除结点时的指针变化情况

注意：在图 8-2～图 8-5 中，①为插入运算前的指针的走向，②为插入运算后的指针的走向；虚线为插入结点后的指针指向。

真题链接

【例 8-1】对于线性表，相对于顺序存储，采用链表存储的缺点是__(57)__。(2014 年 11 月真题 57)

(57) A. 数据元素之间的关系需要占用存储空间，导致存储密度不高

B. 表中结点必须占用地址连续的存储单元，存储密度不高

C. 插入新元素时需要遍历整个链表，运算的时间效率不高

D. 删除元素时需要遍历整个链表，运算的时间效率不高

【解析】链表需要额外的空间存储结点之间的连接关系，因此存储密度不高，但其优点是插入和删除单个元素的时间复杂度为 O(1)。

【答案】A

【例 8-2】若对线性表的最常用操作是访问任意指定序号的元素，并在表尾加入和删除元素，则适宜采用__(57)__存储。(2014 年 5 月真题 57)

(57) A. 顺序表　　B. 单链表　　C. 双向链表　　D. 哈希表

【解析】采用顺序表(即数组)，可以任意访问指定序号的元素，便于在表尾加入和删除元素，但不便于在表头插入和删除元素，在表头操作时需要移动大量元素。要注意的是，题目中要求在表尾加入和删除元素，而不是在表头操作，因此适宜采用顺序表。采用链表插入、删除元素较为方便，但是访问指定序号的元素较为麻烦，需要从头指针开始遍历。

【答案】A

8.1.2 栈和队列

1. 栈

1) 栈的定义及基本运算

栈是只能通过访问它的一端来实现数据存储和检索的一种线性数据结构。栈的修改是按先进后出的原则进行的。因此，栈又称为先进后出(FILO，或后进先出，LIFO)的线性表。对于栈，进行插入和删除操作的一端称为栈顶，另一端称为栈底。不含数据元素的栈称为空栈。

对栈进行的基本操作有如下几种。

- 置空栈 InitStack(S)：创建一个空栈 S。
- 判栈空 Empty(S)：当栈 S 为空栈时返回真值，否则返回假值。
- 入栈 Push(S, x)：将元素 x 加入栈顶，并更新栈顶指针。
- 出栈 Pop(S)：将栈顶元素从栈中删除，并更新栈顶指针。若需要得到栈顶元素的值，可将 Pop(S)定义为一个函数，它返回栈顶元素的值。
- 读栈顶元素 Top(S)：返回栈顶元素的值，但不修改栈顶指针。

2) 栈的存储结构

栈的存储结构有如下两种。

(1) 顺序存储。栈的顺序存储是指用一组地址连续的存储单元依次存储自栈顶到栈底的数据元素，同时附设指针 Top 指示栈顶元素的位置。在顺序存储方式下，需要预先定义或申请栈的存储空间，也就是说栈空间的容量是有限的。因此在顺序栈中，当一个元素入栈时，需要判断是否栈满，若栈满，则元素入栈会发生上溢现象。

(2) 链式存储。用链表作为存储结构的栈也称为链栈。由于栈中元素的插入和删除仅在栈顶一端进行，因此不必设置头结点，链表的头指针就是栈顶指针。

3) 栈的应用

栈的典型应用包括表达式求值、括号匹配等，在计算机语言的实现以及将递归过程转变为非递归过程的处理中，栈有重要的作用。

2. 队列

1) 队列的定义及基本运算

队列是一种先进先出(FIFO)的线性表，它只允许在表的一端插入元素，而在表的另一端删除元素。在队列中，允许插入元素的一端称为队尾(Rear)，允许删除元素的一端称为队头(Front)。

对队列进行的基本操作如下。

(1) 置队空 InitQueue(Q)：创建一个空的队列 Q。

(2) 判队空 Empty(Q)：判断队列是否为空。

(3) 入队 EnQueue(Q, x)：将元素 x 加入到队列 Q 的队尾，并更新队尾指针。

(4) 出队 DeQueue(Q)：将队头元素从队列 Q 中删除，并更新队头指针。

(5) 读队头元素 Frontque(Q)：返回队头元素的值，但并不更新队头指针。

2) 队列的存储结构

队列的存储结构有如下两种。

(1) 顺序存储。队列的顺序存储结构是利用一组地址连续的存储单元存放队列中的元素。由于队列中元素的插入和删除限定在队列的两端进行，因此设置队头指针和队尾指针，分别指示当前的队首元素和队尾元素。

(2) 链式存储。用链表表示的队列简称为链队列。为了便于操作，给链队列添加一个头结点，并令头指针指向头结点。队列为空的判定条件是：头指针和尾指针的值相同，且均指向头结点。

3) 队列的应用

队列结构常用于处理需要排队的场合，如操作系统中处理打印任务的打印队列、离散事件的计算机模拟等。

真题链接

【例 8-3】某双端队列如图 8-6 所示，要求元素进出队列必须在同一端口，即从 A 端进入的元素必须从 A 端出、从 B 端进入的元素必须从 B 端出，则对于 4 个元素的序列 el、e2、e3、e4，若要求前 2 个元素 el、e2 从 A 端口按次序全部进入队列，后两个元素 e3、e4 从 B 端口按次序全部进入队列，则可能得到的出队序列是 (60) 。(2014 年 5 月真题 50)



图 8-6　双端队列

(60) A. el、e2、e3、e4　　B. e2、e3、e4、el

C. e3、e4、el、e2　　D. e4、e3、 e2、el

【解析】el、e2 从 A 端口按次序进入队列，由于从 A 端进入的元素必须从 A 端出，则 e2 要先于 e1 出队；e3、e4 从 B 端口按次序进入队列，而从 B 端进入的元素必须从 B 端出，则 e4 要先于 e3 出队，只有选项 D 满足要求。

【答案】D

【例 8-4】设循环队列 Q 的定义中有 front 和 size 两个域变量，其中 front 表示队头元素的指针，size 表示队列的长度，如图 8-7 所示(队列长度为 3，队头元素为 X、队尾元素为 Z)。设队列的存储空间容量为 M，则队尾元素的指针为__(58)__。(2013 年 11 月真题 58)

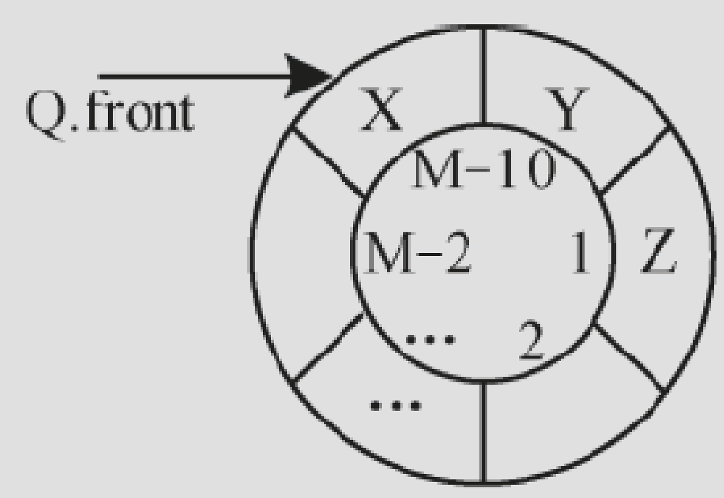


图 8-7　循环队列 Q 示意图

(58) A. (Q.front+Q.size−1)　　B. (Q.front+Q.size −1+M)%M

C. (Q.front−Q.size)　　D. (Q.front − Q.size+M)%M

【解析】考虑到循环，会对 M 进行求模，元素的指针从 0 开始到 M−1，所以队尾元素指针为选项 B。

【答案】B

8.1.3　串

1. 串的定义及基本运算

串是仅由字符构成的有限序列，是取值范围受限的线性表。一般记为 S=‘$a_1a_2\cdots a_n$’，其中 S 是串名，$a_1\ a_2\cdots a_n$ 是串值。下面介绍串的几个基本概念。

(1) 空串：长度为零的串，空串不包含任何字符。

(2) 空格串：由一个或多个空格组成的串。

(3) 子串：由串中任意长度的连续字符构成的序列。含有子串的串称为主串。子串在主串中的位置是指子串首次出现时，该子串的第一个字符在主串中的位置。空串是任意串的子串。

(4) 串相等：指两个串长度相等且对应位置上的字符也相同。

(5) 串比较：两个串比较大小时以字符的 ASCII 码值作为依据。比较操作从两个串的第一个字符开始进行，字符的 ASCII 码值大者所在的串为大；若其中一个串先结束，则以串长较大者为大。

对串进行的基本操作有如下几种。

(1) 赋值操作 StrAssign(s, t)：将串 t 的值赋给串 s。

(2) 连接操作 Concat(s, t)：将串 t 接续在串 s 的尾部，形成一个新串。

(3) 求串长 StrLength(s)：返回串 s 的长度。

(4) 串比较 StrCompare(s, t)：比较两个串的大小。

(5) 求子串 SubString(s, start, len)：返回串 s 中从 start 开始的、长度为 len 的字符序列。

2. 串的存储结构

1) 串的静态存储：定长存储结构

串的顺序存储结构是用一组地址连续的存储单元来存储串值的字符序列。由于串中的元素为字符，因此可通过程序语言提供的字符数组定义串的存储空间，也可以根据串长的需要动态申请字符串的空间。

2) 串的链式存储：块链

串也可采用链表方式作为存储结构，当用链表存储串中的字符时，每个结点中可以存储一个字符，也可以存储多个字符，要考虑存储密度问题。在链式存储结构中，结点的大小的选择和顺序存储方法中数组空间大小的选择一样重要，它直接影响对串处理的效率。

3. 串的模式匹配

子串的定位操作通常称为串的模式匹配，它是各种串处理系统中最重要的运算之一。子串也称为模式串。

1) 朴素的模式匹配算法

朴素的模式匹配算法也称为布鲁特−福斯算法，其基本思想是：从主串的第一个字符起与模式串的第一个字符比较，若相等，则继续逐个字符进行后续的比较，否则从主串的第二个字符起与模式串的第一个字符重新比较，直至模式串中的每个字符依次和主串中的一个连续的字符序列相等，则称匹配成功，否则称匹配失败。

2) 改进的模式匹配算法

改进的模式匹配算法又称为 KMP 算法，其改进之处在于：每当匹配过程中出现相比较的字符不相等时，不需要回溯主串的指针，而是利用已经得到的“部分匹配”的结果，将模式串向后“滑动”尽可能远的距离，再继续进行比较。

真题链接

【例 8-5】在字符串的 KMP 模式匹配算法中，需先求解模式串的 next 函数值，其定义如下式所示，j 表示模式串中字符的序号(从 1 开始)。若模式串 p 为“abaac”，则其 next 函数值为 (60) 。(2014 年 11 月真题 60)

$$\text{next}[j]=\begin{cases}0 & j=1\\ \max\{k\,|\,1<k<j,\text{‘}p_1p_2Lp_{k-1}\text{’}=\text{‘}p_{j-k+1}p_{j-k+2}Lp_{j-1}\text{’}\} & \\ 1 & \text{其他情况}\end{cases}$$

(60) A. 01234　　B. 01122　　C. 01211　　D.01111

【解析】根据公式依次推导即可。

【答案】B

8.2　数组、矩阵和广义表

考核说明：本节主要介绍数组、矩阵和广义表的定义、基本操作和存储结构，存储结构部分是考核要点，要重点掌握。

8.2.1　数组

1. 数组的定义及基本运算

n 维数组是一种“同构”的数据结构，其每个元素类型相同，结构一致。数组是定长线性表在维数上的扩张，即线性表中的元素又是一个线性表。

数组结构的特点是：数据元素数目固定；数据元素具有相同的类型；数据元素的下标关系具有上下界的约束且下标有序。

对数组进行的基本运算有如下两种。

(1)　给定一组下标，存取相应的数据元素。

(2)　给定一组下标，修改相应的数据元素中某个数据项的值。

2. 数组的顺序存储

一旦定义了数组，结构中的数据元素个数和元素之间的关系就不再发生变动，因此数组适合于采用顺序存储结构。

由于计算机的内存结构是一维线性的，因此存储多维数组时必须按照某种方式进行降维处理，即将数组元素排成一个线性序列，这就产生了次序约定问题。对二维数组有两种存储方式：一种是以列为主序的存储方式；另一种是以行为主序的存储方式。

设每个数据元素占用 L 个单元，m、n 为数组的行数和列数，那么以行为主序优先存储的地址计算公式为

$$LOC(a_{ij})= LOC(a_{11})+((i-1)\times n+(j-1))\times L$$

同样，以列为主序优先存储的地址计算公式为

$$LOC(a_{ij})= LOC(a_{11})+((j-1)\times m+(i-1))\times L$$

真题链接

【例 8-6】对二维数组 a[1..N,1..N]中的一个元素 a[i,j]($1\leqslant i,j\leqslant N$)，存储在 a[$i$,$j$]之前的元素个数__(21)__。(2012 年 5 月真题 21)

(21) A. 与按行存储或按列存储方式无关

B. 在 i=j 时与按行存储或按列存储方式无关

C. 在按行存储方式下比按列存储方式下要多

D. 在按行存储方式下比按列存储方式下要少

【解析】存储在a[i,j]之前的元素个数与按行存储或按列存储方式有关。按行存储时，存储在a[i, j]之前的元素个数为$(i-1)*N+j-1=iN+j-N-1$；按列存储时，存储在a[i,j]之前的元素个数为$(j-1)*N+i-1=jN+i-N-1$。很显然，$i<j$时，在按行存储方式下比按列存储方式下要多；$i>j$时，在按行存储方式下比按列存储方式下

要少。

【答案】B

8.2.2　矩阵

1. 特殊矩阵

若矩阵中元素(或非零元素)的分布有一定的规律，则称为特殊矩阵。常见的特殊矩阵有对称矩阵、三角矩阵、对角矩阵等。

对称矩阵：若矩阵 $\boldsymbol{A}_{n\times n}$ 中的元素具有以下特点：

$$a_{ij}=a_{ji}\quad(1\leqslant i，j\leqslant n)$$

则称为 n 阶对称矩阵。

上(下)三角矩阵：矩阵的上(下)三角(不包括对角线)中的元素均为常数或零。

对角矩阵：矩阵中的非零元素都集中在以主对角线为中心的带状区域中，即除了主对角线上和直接在主对角线上、下方若干条对角线上的元素外，其余的矩阵元素都为零。

2. 稀疏矩阵

在一个矩阵中，若非零元素的个数远远少于零元素的个数，且非零元素的分布没有规律，则称为稀疏矩阵。存储稀疏矩阵的非零元素时必须同时存储其位置(行、列号)，用三元组(i, j, a_{ij})可唯一确定矩阵中的一个元素。因此，一个稀疏矩阵可由表示非零元素的三元组及其行、列数唯一确定。

稀疏矩阵的三元组表的顺序存储结构称为三元组顺序表，常用的三元组表的链式存储结构是十字链表。

8.2.3　广义表

1. 广义表的定义

广义表是线性表的推广，是由零个或多个单元素或子表所组成的有限序列。

广义表与线性表的区别在于：线性表的元素都是结构上不可分的单元素，而广义表的元素既可以是单元素，也可以是有结构的表。

广义表一般记为

$$\mathrm{LS}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$$

式中，$\alpha_i(1\leqslant i\leqslant n)$既可以是单个元素，又可以是广义表，分别称为原子和子表。

广义表的长度是指广义表中元素的个数，深度是指广义表展开后所含的括号的最大层数。

2. 广义表的基本操作

下面介绍广义表的基本操作。

(1)　取表头 head(LS)。非空广义表 LS 的第一个元素称为表头，它可以是一个单元素，

也可以是一个子表。

(2) 取表尾 tail(LS)。在非空广义表中，除表头元素之外，由其余元素所构成的表称为表尾。非空广义表的表尾必定是一个表。

3. 广义表的存储结构

广义表通常采用链式存储结构。若广义表不空，则可分解为表头和表尾两部分；反之，一对确定的表头和表尾可唯一决定一个广义表。

8.3 树

考核说明： 本节主要介绍二叉树的定义、运算、性质、存储结构、遍历，线索二叉树，最优二叉树，以及树和森林的转换等内容。二叉树的性质、遍历和哈夫曼树是考核重点，需要掌握相关理论和 C 语言算法描述。

8.3.1 树的定义

树是 $n(n\geqslant0)$个结点的有限集合，n=0 时称为空树，在任一非空树中：

(1) 有且仅有一个称为根的结点。

(2) 其余的结点可分为 $m(m\geqslant0)$个互不相交的子集 $T_1, T_2, \cdots, T_m$，其中每个子集本身又是一棵树，并称其为根结点的子树。

树的递归定义表明了树的固有特性，也就是一棵树由若干棵子树构成，而子树又由更小的子树构成。

树中的基本概念如下。

(1) 双亲和孩子：结点的子树的根称为该结点的孩子；该结点称为其子结点的双亲。

(2) 兄弟：具有相同双亲的结点互为兄弟。

(3) 结点的度：一个结点的子树的个数记为该结点的度。

(4) 叶子结点：也称为终端结点，指度为零的结点。

(5) 内部结点：度不为零的结点称为分支结点或非终端结点。除根结点之外，分支结点也称为内部结点。

(6) 结点的层次：根为第一层，根的孩子为第二层，以此类推。

(7) 树的高度：一棵树的最大层次数记为树的高度(或深度)。

(8) 有序(无序)树：若将树中的结点的各子树看成从左到右具有次序的，即不能交换，则称该树为有序树，否则称为无序树。

(9) 森林：是 $m(m\geqslant0)$棵互不相交的树的集合。

8.3.2　二叉树

1. 二叉树的定义

二叉树(Binary Tree)是 $n(n\geqslant 0)$个结点的有限集合，它或者是空树(n=0)，或者是由一个根结点及两棵互不相交的、分别称为左子树和右子树的二叉树所组成。

二叉树与树的区别如下。

- 二叉树的结点的子树要区分左子树和右子树，即使在结点只有一棵子树的情况下也要明确指出该子树是左子树还是右子树。
- 二叉树的结点的最大度数为 2，树中不限制结点的度数。

2. 二叉树的运算

二叉树的基本运算是遍历，其他运算可建立在遍历运算的基础上。

3. 二叉树的性质

二叉树具有如下性质。

(1)　二叉树第 i 层上的结点数目最多为 $2^{i-1}(i\geqslant 1)$。

(2)　深度为 k 的二叉树至多有 2^k-1 个结点($k\geqslant 1$)。

(3)　在任意一棵二叉树中，若终端结点数为 n_0，度为 2 的结点数为 n_2，则 $n_0=n_2+1$。

(4)　具有 n 个结点的完全二叉树的深度为$\lfloor \log_2 n \rfloor+1$。

(5)　对一棵有 n 个结点的完全二叉树的结点按层次自左至右进行编号，则对任一结点 i 有：

- 若 i=1，则结点 i 是二叉树的根，无双亲；若 i>1，则其双亲为$\left\lfloor \frac{i}{2} \right\rfloor$。
- 若 $2i>n$，则结点 i 无左孩子，否则其左孩子为 $2i$。
- 若 $2i+1>n$，则结点 i 无右孩子，否则其右孩子为 $2i+1$。

若深度为 k 的二叉树有 2^k-1 个结点，则称其为满二叉树。

深度为 k、有 n 个结点的二叉树，当且仅当其每一个结点都与深度为 k 的满二叉树编号从 1 至 n 的结点一一对应时，称为完全二叉树。

4. 二叉树的存储结构

1)　顺序存储结构

用一组地址连续的存储单元存储二叉树中的数据元素，必须把结点排成一个适当的线性序列，并且结点在这个序列中的相互位置能反映出结点之间的逻辑关系。

2)　链式存储结构

由于二叉树中的结点包含数据元素、左子树根、右子树根及双亲等信息，因此可以用三叉链表或二叉链表来存储二叉树，链表的头指针指向二叉树的根结点。

5. 二叉树的遍历

遍历是按某种策略访问树中的每个结点，且仅访问一次。由于二叉树所具有的递归性质，一棵非空的二叉树可以看作由根结点、左子树和右子树三部分构成，因此若能依次遍历这三个部分中的每个结点信息，也就遍历了整棵二叉树。按照遍历左子树要在遍历右子树之前进行的约定，根据访问根结点位置的不同，可得到二叉树的前序、中序和后序三种遍历方法。

遍历二叉树的基本操作就是访问结点，不论按照哪种次序遍历，对含有 n 个结点的二叉树，遍历算法的时间复杂度都为 $O(n)$。在最坏情况下，二叉树是有 n 个结点且深度为 n 的单枝树，遍历算法的空间复杂度也为 $O(n)$。

对二叉树还可以进行层序遍历。层序遍历就是从树的根结点出发，首先访问第 1 层的根结点，然后从左到右依次访问第 2 层上的结点，以此类推，自上而下、自左到右逐层访问树中各层上结点的过程就是层序遍历。

6. 线索二叉树

若 n 个结点的二叉树采用链表作存储结构，则链表中含有 n+1 个空指针域，利用这些空指针域来存放指向结点的前驱和后继信息。线索链表的结点结构如图 8-8 所示。

ltag	lchild	data	rchild	rtag

图 8-8　线索链表的结点结构

若二叉树的二叉链表采用图 8-8 所示的结点结构，则相应的链表称为线索链表，其中指向结点前驱、后继的指针称为线索，加上线索的二叉树称为线索二叉树。对二叉树以某种次序遍历使其变为线索二叉树的过程称为线索化。

7. 二叉树的应用：最优二叉树

哈夫曼树又称最优二叉树，是一类带权路径长度最短的树。

路径：是指从树中一个结点到另一个结点之间的通路，路径上的分支数目称为路径长度。

树的路径长度：是从树根到每一个叶子的路径长度之和。结点的带权路径长度为从该结点到树根之间的路径长度与该结点权的乘积。

树的带权路径长度：指树中所有叶子结点的带权路径长度之和，记为

$$\mathrm{WPL}=\sum_{i=1}^{n} w_i l_i$$

式中：n 为带权叶子结点的数目；w_i 为叶子结点的权值；l_i 为叶子结点到根的路径长度。

哈夫曼树：是指权值为 $w_1, w_2, \cdots, w_n$ 的 n 个叶子结点的二叉树中带权路径长度最小的二叉树。

构造最优二叉树的哈夫曼算法如下。

(1) 根据给定的 n 个权值 $w_1, w_2, \cdots, w_n$ 构成 n 棵二叉树的集合 $F=\{T_1, T_2, \cdots, T_n\}$，其中每棵二叉树 T_i 中只有一个带权为 w_i 的根结点，其左右子树均空。

(2) 在 F 中选取两棵根结点的权值最小的树作为左右子树，构造一棵新的二叉树，置

新构造二叉树的根结点的权值为其左右子树根结点的权值之和。

(3) 从 F 中删除这两棵树，同时将新得到的二叉树加入到 F 中。

重复(2)(3)，直到 F 中只含一棵树时为止。这棵树便是哈夫曼树。

真题链接

【例 8-7】在某个二叉查找树(即二叉排序树)中进行查找时，效率最差的情形是该二叉查找树是__(59)__。(2014 年 11 月真题 59)

(59) A. 完全二叉树　　B. 平衡二叉树　　C. 单枝树　　D. 满二叉树

【解析】单枝树极度不平衡，查找的平均时间复杂度为$O(n)$。

【答案】C

【例 8-8】某二叉树如图 8-9 所示，若进行顺序存储(即用一维数组元素存储该二叉树中的结点且通过下标反映结点间的关系，例如，对于下标为 i 的结点，其左孩子的下标为 $2i$、右孩子的下标为 $2i+1$)，则该数组的大小至少为__(58)__；若采用三叉链表存储该二叉树(各个结点包括结点的数据、父结点指针、左孩子指针、右孩子指针)，则该链表的所有结点中空指针的数目为__(59)__。(2014 年 5 月真题 58～59)

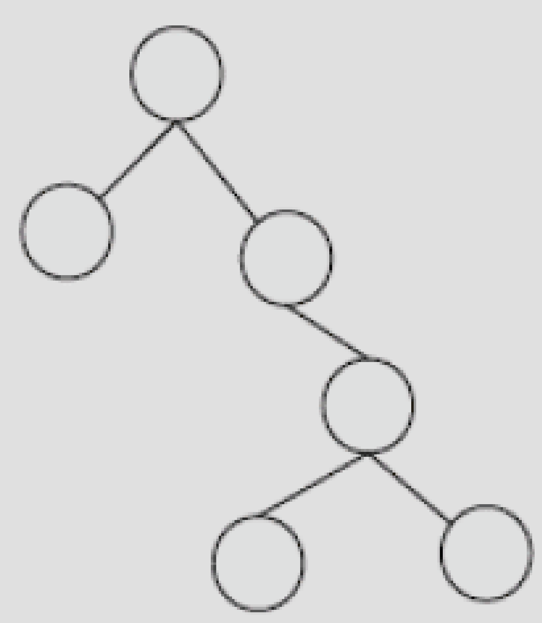


图 8-9　二叉树

(58) A.6　　B.10　　C.12　　D.15

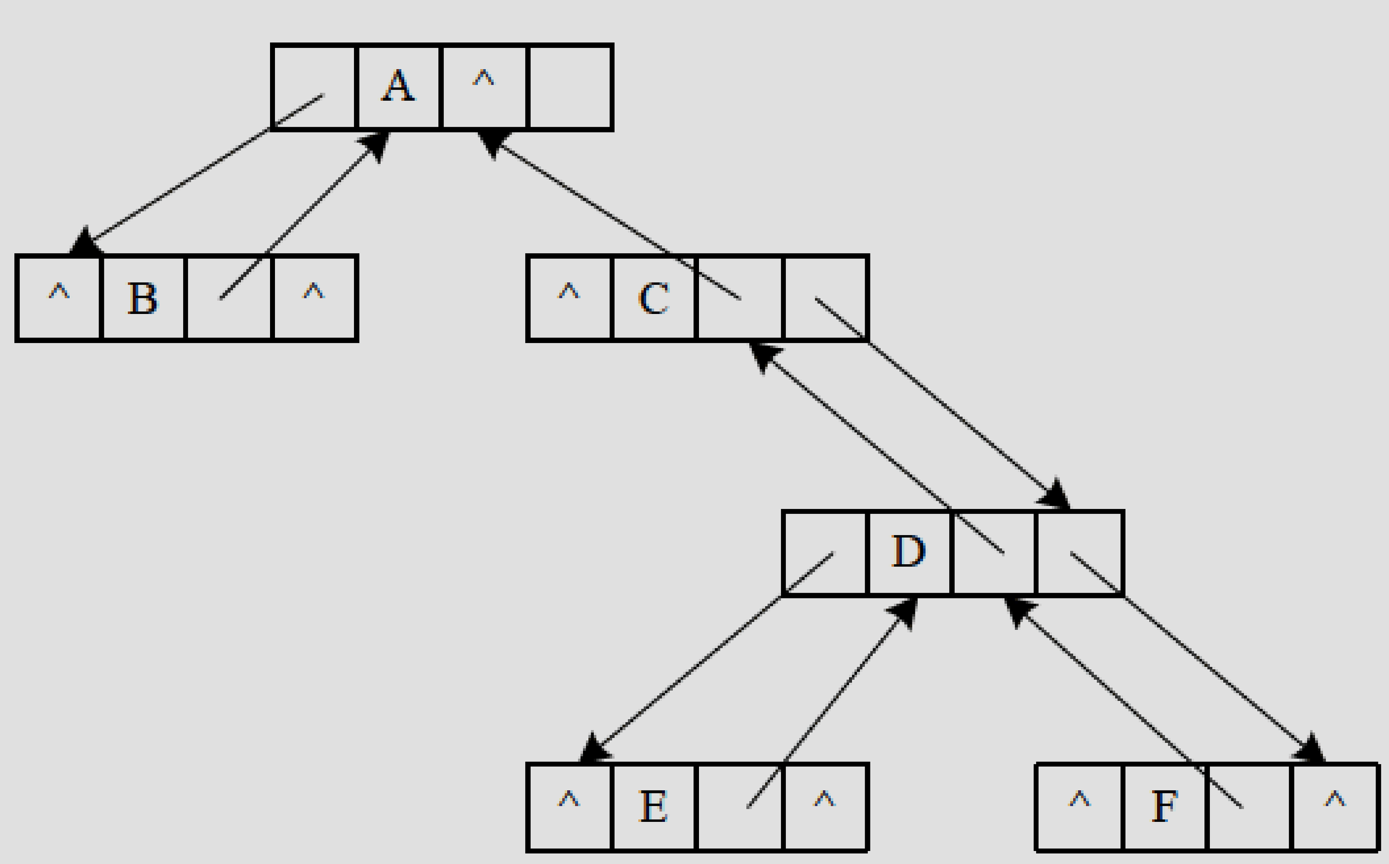
(59) A.6　　B.8　　C.12　　D.14

【解析】采用顺序存储结构存储二叉树时，一般的二叉树也必须按照完全二叉树的形式存储，需要填上一些不存在的“虚结点”。题中二叉树的高度为 4，需要的存储空间为 $2^4-1=15$，如下：

A	B	C	^	^	^	D	^	^	^	^	^	^	E	F

采用三叉链表存储二叉树，如下：

左孩子	结点数据	父结点	右孩子



可见，空指针的数目为 8。

【答案】(58)D；(59)B

【例 8-9】已知一个文件中出现的各个字符及其对应的频率如表 8-1 所示。若采用定长编码，则该文件中字符的码长应为__(64)__。若采用 Huffman 编码，则字符序列“face”的编码应为__(65)__。(2014 年 11 月真题 64～65)

表 8-1　字符及其对应的频率

字符	a	b	c	d	e	f
频率(%)	45	13	12	16	9	5

(64)　A. 2　　B. 3　　C. 4　　D. 5

(65)　A. 110001001101　　B. 001110110011　　C. 101000010100　　D. 010111101011

【解析】(64)有 6 个不同字母，需要采用 3 位二进制进行编码。

(65)哈夫曼静态编码：它对需要编码的数据进行两遍扫描：第一遍统计原数据中各字符出现的频率，利用得到的频率值创建哈夫曼树，并必须把树的信息保存起来，即把字符 0～255(2^8=256)的频率值以 2～4 字节的长度顺序存储起来(用 4 字节的长度存储频率值，频率值的表示范围为 0～$2^{32}-1$，这已足够表示大文件中字符出现的频率了)，以便解压时创建同样的哈夫曼树进行解压；第二遍则根据第一遍扫描得到的哈夫曼树进行编码，并把编码后得到的码字存储起来。

【答案】(64)B；(65)A

8.4　图

考核说明：本节主要介绍图的定义、存储结构和遍历，生成树和最小生成树，拓扑排序和关键路径，以及最短路径等内容，考试中会出现少量的题目。图的存储结构和遍历算法相对来说较为重要，要能理解掌握。

8.4.1　图的定义

图 G 是由两个集合 V 和 E 构成的二元组，记作 $G=(V,E)$，其中 V 是图中顶点的非空有限集合，E 是图中边的有限集合。从数据结构的逻辑关系来看，图中任一顶点都有可能与图中其他顶点有关系，而图中所有顶点都有可能与某一顶点有关系。在图中，数据结构中的数据元素用顶点表示，数据元素之间的关系用边表示。

(1)　有向图：若图中每条边都是有方向的，则称 G 为有向图。顶点间的关系用 $<v_i,v_j>$ 表示，它说明从 v_i 到 v_j 的一条有向边(也称为弧)。v_i 是有向边的起点，称为弧尾；v_j 是有向边的终点，称为弧头。

(2)　无向图：若图中的每条边都是无方向的，则顶点 v_i 和 v_j 之间的边用 (v_i,v_j) 表示。

(3)　无向完全图：若一个无向图具有 n 个顶点，而每一个顶点与其他 $n-1$ 个顶点之间都有边，则称为无向完全图。显然，含有 n 个顶点的无向完全图共有 $n(n-1)/2$ 条边。

(4) 有向完全图：有 n 个顶点的有向完全图中弧的数目为 $n(n-1)$，即任何两个不同顶点之间都有方向相反的两条弧存在。

(5) 度、入度和出度：顶点的度是指关联于该顶点的边的数目，记为 $D(v)$。若 G 为有向图，顶点的度表示该顶点的入度和出度之和。顶点的入度是指以该顶点为终点的有向边的数目，而顶点的出度是指以该顶点为起点的有向边的数目，分别记为 ID(v)和 OD(v)。无论有向图还是无向图，顶点数 n、边数 e 与各顶点的度之间有以下关系：

$$e=\frac{1}{2}\sum_{i=1}^{n}D(v_i)$$

(6) 路径：在无向图 G 中，从顶点 v_p 到顶点 v_q 的路径是指存在一个顶点序列 v_p，$v_{i1},v_{i2},\cdots,v_{in}$，$v_q$，使得($v_{i1},v_p$)，($v_{i1},v_{i2}$)，…，($v_{in},v_q$)均属于 $E(G)$。

(7) 子图：对于两个图 $G=(V,E)$和 $G'=(V',E')$，如果 V'是 V 的子集，E'是 E 的子集，则称 G'为 G 的子图。

(8) 连通图：在无向图 G 中，若从顶点 v_i 到顶点 v_j 有路径，则称顶点 v_i 和顶点 v_j 是连通的。如果无向图 G 中任意两个顶点都是连通的，则称其为连通图。无向图 G 的极大连通子图称为 G 的连通分量。

(9) 强连通图：在有向图 G 中，如果对于每一对顶点 v_i,v_j 且 $v_i\neq v_j$，从顶点 v_i 到顶点 v_j 和从顶点 v_j 到顶点 v_i 都存在路径，则称图 G 为强连通图。$V(G)$中任意两个不同的顶点 v_i 和 v_j，都存在从 v_i 到 v_j 以及从 v_j 到 v_i 的路径，则称 G 是强连通图。

(10) 网：边(或弧)带权值的图称为网。

(11) 生成树：一个连通图的生成树是一个极小的连通子图，它包含图中的全部顶点，但只有构成一棵树的 $n-1$ 条边。

(12) 有向树和生成森林：如果一个有向图恰有一个顶点的入度为 0，其余顶点的入度均为 1，则是一棵有向树。有向图的生成森林由若干棵有向树组成，含有图中全部顶点，但只有足以构成若干棵不相交的有向树的弧。

真题链接

【例 8-10】在一个有向图 G 的拓扑序列中，顶点 v_i 排列在 v_j 之前，说明图 G 中__(59)__。(2013 年 11 月真题 59)

(59) A. 一定存在弧 $<v_i,v_j>$

B. 一定存在弧 $<v_j,v_i>$

C. 可能存在 v_i 到 v_j 的路径，而不可能存在 v_j 到 v_i 的路径

D. 可能存在 v_j 到 v_i 的路径，而不可能存在 v_i 到 v_j 的路径

【解析】根据有向图 G 的拓扑序列定义，顶点 v_i 排列在 v_j 之前，可以得知可能存在 v_i 到 v_j 的路径，拓扑序列是单向的，所以不可能从 v_j 到 v_i 的路径。所以本题答案选 C。

【答案】C

8.4.2 存储结构

1. 邻接矩阵表示法

对于具有 n 个顶点的图 $G(V, E)$来说，其邻接矩阵是一个 n 阶方阵，且满足

$$A[i][j]=\begin{cases}1, \text{若}(v_i,v_j)\text{或}<v_i,v_j>\text{是}E\text{中的边}\\0, \text{若}(v_i,v_j)\text{或}<v_i,v_j>\text{不是}E\text{中的边}\end{cases}$$

由邻接矩阵的定义可知，无向图的邻接矩阵是对称的，有向图的邻接矩阵就不一定对称了。借助于邻接矩阵易判定任意两个顶点之间是否有边(或弧)相连，并且容易求得各个顶点的度。

网(赋权图)的邻接矩阵可定义为

$$A[i][j]=\begin{cases}W_{ij}, \text{若}(v_i,v_j)\text{或}<v_i,v_j>\text{是}E\text{中的边}\\\infty, \text{若}(v_i,v_j)\text{或}<v_i,v_j>\text{不是}E\text{中的边}\end{cases}$$

2. 邻接链表表示法

邻接链表指的是：为图的每个顶点建立一个单链表，第 i 个单链表中的结点表示依附于顶点 v_i 的边(对于有向图是以 v_i 为尾的弧)。邻接链表中的结点有表结点和表头结点两种类型。

邻接矩阵和邻接链表表示法对有向图和无向图都适用。

8.4.3 图的遍历

1. 深度优先遍历

从图 G 中任意一个顶点 v 出发，深度优先遍历(DFS)的算法步骤如下。

(1) 设立搜索指针 p，使 p 指向顶点 v。

(2) 访问 p 顶点，并使 p 指向与 p 顶点相邻接的且尚未被访问过的顶点。

(3) 若 p 不空，则重复步骤(3)，否则执行步骤(4)。

(4) 沿着刚才访问的次序、方向回溯到一个尚有邻接顶点且未被访问过的顶点，并使 p 指向这个未被访问的邻接顶点，然后重复步骤(2)，直至所有的顶点均被访问为止。

这个算法的特点是尽可能先对纵深方向搜索，因此可以很容易得到其遍历的递归算法。

2. 广度优先遍历

广度优先遍历(BFS)的遍历过程是：假设从图中某一个顶点 v 出发，在访问 v 之后依次访问 v 的各个未被访问过的邻接点，然后分别从这些邻接点出发依次访问它们的邻接点，并使“先被访问的顶点的邻接点”先于“后被访问的顶点的邻接点”被访问，直至图中所有已被访问过的顶点的邻接点都被访问到。若此时还有未被访问的顶点，则另选其中一个作为起点，重复上述过程，直至图中所有的顶点都被访问到为止。

广度优先遍历图的特点是尽可能先进行横向搜索，即最先访问的顶点的邻接点亦先被访问。

8.4.4　生成树和最小生成树

1. 生成树

设图 $G=(V, E)$是个连通图，当从图中任意一个顶点出发遍历图 G 时，将边集 $E(G)$分为两个集合 $A(G)$和 $B(G)$。其中 $A(G)$是遍历时所经过的边的集合，$B(G)$是遍历时未经过的边的集合。$G_1=(V, A)$是图 G 的子图，称子图 G_1 为连通图 G 的生成树。

2. 最小生成树

对于连通图来说，边是带权值的，生成树的各边也带权值，如果把生成树各边的权值总和称为生成树的权，则把权值最小的生成树称为最小生成树。

构造生成树有多种算法，其中多数算法利用了最小生成树的 MST 性质：假设 $G=(V, E)$是一个连通图，U 是顶点集 V 的一个非空子集。若(u,v)是一条最小权值的边，其中 $u \in U, v \in V-U$，则必存在一棵包含边(u,v)的最小生成树。

8.4.5　拓扑排序和关键路径

1. AOV 网

在有向图中，若一顶点表示活动，用有向边表示活动之间的优先关系，则称这样的有向图为以顶点表示活动的网，简称 AOV 网。

在 AOV 网中不应出现有向环。不存在回路的 AOV 网称为有向无环图，或 DAG 图。检测的方法是对有向图构造其顶点的拓扑有序序列，若图中所有顶点都在它的拓扑有序序列中，则该 AOV 网中必定不存在环。

2. 拓扑排序及其算法

拓扑排序是将 AOV 网中所有顶点排成一个线性序列，该序列满足：若在 AOV 网中从顶点 v_i 到 v_j 有一条路径，则在该线性序列中，顶点 v_i 必然在顶点 v_j 之前。拓扑排序即指对 AOV 网构造拓扑序列的操作。

对 AOV 网进行拓扑排序的方法如下。

(1) 在 AOV 网中选择一个入度为零的顶点且输出它。

(2) 从网中删除该顶点及与该顶点有关的所有边。

(3) 重复上述两步，直至网中不存在入度为零的顶点为止。

若在 AOV 网中考察各顶点的出度，并按下列步骤进行排序，则称为逆拓扑排序。

(1) 在 AOV 网中选择一个没有后继的顶点且输出它。

(2) 从网中删除该顶点，并删去所有到达该顶点的弧。

(3) 重复上述两步，直至网中不存在出度为零的顶点为止。

拓扑排序的时间复杂度为$O(n+e)$。

3. AOE 网

若在带权有向图 G 中以顶点表示事件，以有向边表示活动，边上的权值表示该活动持续的时间，则这种带权有向图称为用边表示活动的网，简称 AOE 网。AOE 网中不应存在有向回路。

8.4.6 最短路径

1. 单源点最短路径

单源点最短路径是指：给定带权有向图 G 和源点 v，求从 v 到 G 中其余各顶点的最短路径。迪杰斯特拉(Dijkstra)提出了按路径长度递增的次序产生最短路径的算法。

2. 每对顶点间的最短路径

若每次以一个顶点为源点，重复执行迪杰斯特拉算法 n 次，便可求得网中每一对顶点之间的最短路径。弗洛伊德(Floyd)提出了求最短路径的算法，该算法在形式上要简单一些。

8.5 查　　找

考核说明：本节主要介绍查找的基本概念、静态查找表、动态查找表、哈希表等内容，考试中经常考到，要掌握各种查找算法的基本思想和 C 语言描述。

8.5.1 查找的基本概念

1. 基本概念

查找是一种常用的基本运算。查找表是指由同一类型的数据元素构成的集合。

- 静态查找表：对查找表经常要进行的两种操作是查询和检索。
- 动态查找表：对查找表经常要进行的操作是插入和删除。
- 关键字：是数据元素的某个数据项的值，用它来识别这个数据元素。
- 主关键字：能唯一标识一个数据元素的关键字。
- 次关键字：能标识多个数据元素的关键字。
- 查找：根据给定的某个值，在查找表中确定是否存在一个其关键字等于给定值的记录或数据元素的过程称为查找。

2. 查找操作的性能分析

通常以“其关键字和给定值进行过比较的记录个数的平均值”作为衡量查找算法好坏的依据。

平均查找长度：为确定记录在查找表中的位置，须与给定关键字值进行比较的次数的

期望值称为查找算法在查找成功时的平均查找长度。

对于含有 n 个记录的表，平均查找长度(ASL)定义为

$$\mathrm{ASL}=\sum_{i=1}^{n}p_i c_i$$

式中：p_i 为对表中第 i 个记录进行查找的概率，且 $\sum_{i=1}^{n}p_i=1$。一般情况下，认为查找每个记录的概率是相等的，即 $p_i=1/n$。

8.5.2　静态查找表

1. 顺序查找

顺序查找的基本思想是：从表的一端开始，逐个进行记录的关键字和给定值的比较，若找到一个记录的关键字与给定值相等，则查找成功；若整个表中的记录均比较过，仍未找到关键字等于给定值的记录，则查找失败。

顺序查找的性能分析如下。

一般情况下，$c_i=n-i+1$，因此在等概率情况下，顺序查找成功的平均查找长度为

$$\mathrm{ASL}_{\mathrm{ss}}=\sum_{i=1}^{n}p_i c_i=\frac{1}{n}\sum_{i=1}^{n}(n-i+1)=\frac{n+1}{2}$$

也就是说，成功查找的平均次数约为表长的一半。与其他方法相比，顺序查找方法在 n 值较大时，其平均查找长度较大，查找效率较低。但这种方法也有优点，就是算法简单且适应面广，对查找表的结构没有要求，无论记录是否按关键字有序排列均可使用。

2. 折半查找

折半查找又称二分查找，其基本思想是：设查找表的元素存储在一难数组 $r[1..n]$ 中，那么在表中的元素已经按关键字递增(或递减)的方式排序的情况下，可进行折半查找。其方法是：首先将待查的 key 值与表 r 中间位置上(下标为 mid)的记录的关键字进行比较，若相等，则查找成功；若 key>r[mid].key，则说明待查记录只可能在后半个子表 $r[\mathrm{mid}+1..n]$ 中，下一步应在后半个子表中再进行折半查找，若 key<r[mid].key，说明待查记录只可能在前半个子表 $r[1..\mathrm{mid}-1]$ 中，下一步应在 r 的前半个子表中进行折半查找。这样通过逐步缩小范围，直到查找成功或子表为空时失败为止。

折半查找的性能分析：折半查找的过程可以用一棵二叉树描述，设结点总数为 $n=2^h-1$，则判定树是深度为 $h=\log_2(n+1)$ 的满二叉树。在等概率情况下，折半查找的平均查找长度为 $\mathrm{ASL}_{\mathrm{bs}}=\sum_{i=1}^{n}p_i c_i=\frac{1}{n}\sum_{j=1}^{n}j\times 2^{j-1}=\frac{n+1}{n}\log_2(n+1)-1$。

当 n 值较大时，$\mathrm{ASL}_{\mathrm{bs}}\approx\log_2(n+1)-1$。

折半查找比顺序查找的效率要高，但它要求查找表进行顺序存储并且按关键字有序排列，因此对表进行元素的插入和删除时，需要移动大量的元素，所以折半查找适用于表不易变动，且又经常进行查找的情况。

3. 分块查找

分块查找又称索引顺序查找，是对顺序查找方法的一种改进，其性能介于顺序查找和折半查找之间。

分块查找的基本思想是：在分块查找过程中，首先把表分成若干块，每一块中的关键字不一定有序，但块之间是有序的，即后一块中所有记录的关键字均大于前一个块中最大的关键字；此外，还建立了一个索引表，索引表按关键字有序。所以分块查找的过程分为两步：第一步在索引表中确定待查记录所在的块；第二步在块内顺序查找。

真题链接

【例 8-11】实现二分查找(折半查找)时，要求查找表 (61) 。(2014 年 5 月真题 61)

(61) A. 顺序存储，关键码无序排列　　B. 顺序存储，关键码有序排列

C. 双向链表存储，关键码无序排列　　D. 双向链表存储，关键码有序排列

【解析】二分查找又称折半查找，优点是比较次数少，查找速度快，平均性能好；其缺点是要求待查表为有序表，且插入删除困难。因此，折半查找方法适用于不经常变动而查找频繁的有序列表。

二分查找算法要求：①必须采用顺序存储结构；②必须按关键字大小有序排列。

【答案】B

8.5.3 动态查找表

1. 二叉排序树

二叉排序树又称二叉查找树，它或者是一棵空树，或者是满足如下性质的二叉树。

(1) 若它的左子树非空，则左子树上所有结点的值均小于根结点的值。

(2) 若它的右子树非空，则右子树上所有结点的值均大于根结点的值。

(3) 左、右子树本身就是两棵二叉排序树。

二叉排序树的查找过程是：若二叉排序树非空，则将给定值与根结点的关键字值相比较，若相等，则查找成功；若不等，则当根结点的关键字值大于给定值时，到根的左子树中进行查找，否则到根的右子树中进行查找。若找到，则查找过程是走了一条从树根到所找到结点的路径；否则查找过程终止于一棵空树。

2. 平衡二叉树

平衡二叉树又称为 AVL 树，它或者是一棵空树，或者是具有下列性质的二叉树：它的左右子树都是平衡二叉树，且左子树和右子树的深度之差的绝对值不超过 1；若将二叉树结点的平衡因子定义为该结点的左子树的深度减去其右子树的深度，则平衡树上所有结点的平衡因子只可能是-1、0 和 1；只要树上有一个结点的平衡因子的绝对值大于 1，则该二叉树就是不平衡的。

3. B-树

B-树的定义：一棵 m 阶的 B-树，或为空树，或为满足下列特性的 m 叉树。

(1) 树中每个结点至多有 m 棵子树。

(2) 若根结点不是叶子结点，则至少有两棵子树。

(3) 除根之外的所有非终端结点至少有 $\left\lceil \frac{m}{2} \right\rceil$ 棵子树。

(4) 所有的非终端结点中包含下列数据信息：

$$(n, A_0, K_1, A_1, K_2, A_2, \cdots, K_n, A_n)$$

式中：K_i (i=1, 2, …, n)为关键字，且 $K_i < K_{i+1}$ (i=1, 2, …, n−1)；A_i (i=0, 1, 2, …, n)为指向子树根结点的指针，且指针 A_{i-1} 所指子树中所有结点的关键字均小于 K_i (i=1, 2, …, n)，A_n 所指子树中所有结点的关键字均大于 K_n，$n\left(\left\lceil \frac{m}{2} \right\rceil - 1 \leqslant n \leqslant m-1\right)$为结点中关键字的个数。

(5) 所有的叶子结点都出现在同一层次上，并且不带信息(可以看作外部结点或查找失败的结点，实际上这些结点不存在，指向这些结点的指针为空)。

在 B-树上进行查找的过程是：首先在根结点所包含的关键字中查找给定的关键字，若找到则成功返回；否则确定待查找的关键字所在的子树并继续进行查找，直到查找成功或查找失败(指针为空)时为止。

8.5.4 哈希表及其查找

1. 定义

根据设定的哈希函数和处理冲突的方法，将一组关键字映射到一个有限的连续的地址集(区间)上，并以关键字在地址集中的“像”作为记录在表中的存储位置，这种表称为哈希表，这一映射过程称为哈希造表或散列，所得的存储位置称为哈希地址或散列地址。

对于哈希表，主要考虑两个问题：其一是如何构造哈希函数；其二是如何解决冲突。

2. 哈希函数的构造方法

常用的哈希函数构造方法有直接定址法、数字分析法、平方取中法、折叠法、随机数法和除留余数法等。

3. 处理冲突的方法

常见的处理冲突的方法有开放地址法、链地址法、再哈希法、建立一个公共溢出区。

4. 哈希表的查找及其性能分析

从哈希表的查找过程可知以下两点。

- 虽然哈希表在关键字与记录的存储位置之间建立了直接映像，但由于冲突的产生，使得哈希表的查找过程仍然是一个给定值和关键字进行比较的过程，因此，仍需以平均查找长度衡量哈希表的查找效率。
- 查找过程中需与给定值进行比较的关键字的个数取决于哈希函数、处理冲突的方法和哈希表的装填因子三个因素。哈希表的装填因子定义为

$$\alpha = \frac{\text{表中装入的记录数}}{\text{哈希表的长度}}$$

式中 α 标志哈希表的装满程度。直观地看，α 越小，发生冲突的可能性就越小；反之，α 越大，表中已填入的记录越多，再填记录时，发生冲突的可能性就越大，则查找时，给定值需与之进行比较的关键字的个数也就越多。

8.6 排　　序

考核说明： 本节主要介绍排序的基本概念以及各种排序算法，包括简单排序、希尔排序、快速排序、堆排序、归并排序、基数排序和外部排序，是考核重点，下午考试中经常考查排序算法，需要掌握各种排序算法的基本思想和 C 语言描述，能够分析算法的复杂度。

8.6.1 排序的基本概念

排序：假设含 n 个记录的文件内容为 $\{R_1,R_2,\cdots,R_n\}$，其相应的关键字分别为 $\{K_1,K_2,\cdots,K_n\}$。经过排序确定一种排列 $R_{i1},R_{i2},\cdots,R_{in}$，使得它们的关键字满足关系 $K_{i1}\leqslant K_{i2}\leqslant\cdots\leqslant K_{in}$(或 $K_{i1}\geqslant K_{i2}\geqslant\cdots\geqslant K_{in}$)，这样的运算称为排序。

内部排序：指待排序记录全部存放在内存中排序的过程。

外部排序：指待排序记录的数量很大，以致内存不能容纳全部记录，在排序过程中尚需对外存进行访问的过程。

8.6.2 简单排序

下面介绍几种简单排序。

- 直接插入排序：在插入第 i 个记录时，$R_1,R_2,\cdots,R_{i-1}$ 已经排好序，这时将关键字 k_i 依次与关键字 $k_{i-1},k_{i-2},\cdots,k_1$ 进行比较，从而找到应该插入的位置，然后将 k_i 插入，插入位置及其后的记录依次向后移动。
- 冒泡排序：首先将第一个记录的关键字和第二个记录的关键字进行比较，若为逆序，则交换两个记录的值，然后比较第二个记录和第三个记录的关键字，以此类推，直至第 $n-1$ 个记录和第 n 个记录的关键字进行过比较为止。上述过程称作第一趟冒泡排序，其结果是关键字最大的记录被安置到第 n 个记录的位置上，然后进行第二趟冒泡排序，对前 $n-1$ 个记录进行同样的操作，其结果是关键字次大的记录被安置到第 $n-1$ 个记录的位置上，当进行完第 $n-1$ 趟时，所有记录有序排列。
- 简单选择排序：通过 $n-1$ 次关键字之间的比较，从 $n-i+1$ 个记录中选出关键字最小的记录，并和第 i 个记录进行交换，当 i 等于 n 时所有记录有序排列。

真题链接

【例 8-12】对一待排序序列分别进行直接插入排序和简单选择排序，若待排序序列中有两个元素的值相同，则 (63) 保证这两个元素在排序前后的相对位置不变。(2014 年 11 月真题 63)

(63) A. 直接插入排序和简单选择排序都可以

B. 直接插入排序和简单选择排序都不能

C. 只有直接插入排序可以

D. 只有简单选择排序可以

【解析】直接插入排序的工作原理是通过构建有序序列，每次从无序序列中取出一个元素，在已排序序列中从后向前扫描，找到相应位置并插入。因为元素扫描顺序固定，所以对于相同元素，可以保证后出现的元素一定在之前出现元素的后面。

【答案】C

8.6.3　希尔排序

希尔排序又称为缩小增量排序，是对直接插入排序方法的改进。

希尔排序的基本思想是：先将整个待排记录序列分割成若干个子序列，然后分别进行直接插入排序，待整个序列中的记录基本有序时，再对全体记录进行一次直接插入排序。具体做法是：先取定一个小于 n 的整数 d_1 作为第一个增量，把文件的全部记录分成 d_1 个组，将所有距离为 d_1 倍数的记录放在同一个组中，在各组内进行直接插入排序；然后取第二个增量 $d_2 < d_1$，重复上述分组和排序工作，依次类推，直至所取的增量 $d_i = 1(d_i < d_{i-1} < \cdots < d_2 < d_1)$，即所有记录放在同一组进行直接插入排序为止。

8.6.4　快速排序

快速排序的基本思想是：通过一趟排序将待排的记录分割为独立的两部分，其中一部分记录的关键字均比另一部分记录的关键字小，然后再分别对这两部分记录进行排序，以达到整个序列有序。

具体做法是：附设两个指针 low 和 high，它们的初值分别指向文件的第一个记录和最后一个记录。设枢轴记录的关键字为 Pivotkey，则首先从 high 所指位置起向前搜索，找到第一个关键字小于 Pivotkey 的记录并与枢轴记录互相交换；然后从 low 所指位置起向后搜索，找到第一个关键字大于 Pivotkey 的记录并与枢轴记录相互交换；重复这两步直至 low=high 为止。

在所有同数量级 ($O(n\log_2 n)$) 的排序方法中，快速排序被认为是平均性能最好的一种，但是，若初始记录序列按关键字有序或基本有序时，快速排序将退化为冒泡排序，此时算法的时间复杂度为 $O(n^2)$。

真题链接

【例 8-13】快速排序算法在排序过程中，在待排序数组中确定一个元素为基准元素，根据基准元素把

待排序数组划分成两个部分，前面一部分元素值小于基准元素，而后面一部分元素值大于基准元素。然后再分别对前后两个部分进行划分。根据上述描述，快速排序算法采用了 (61) 算法设计策略。已知确定基准元素操作的时间复杂度为 $\Theta(n)$，则快速排序算法的最好和最坏情况下的时间复杂度为 (62) 。(2014 年 11 月真题 61～62)

(61) A. 分治　B. 动态规划　C. 贪心　D. 回溯

(62) A. $\Theta(n)$ 和 $\Theta(n\log_2 n)$　B. $\Theta(n)$ 和 $\Theta(n^2)$

C. $\Theta(n\lg n)$ 和 $\Theta(n\log_2 n)$　D. $\Theta(n\lg n)$ 和 $\Theta(n^2)$

【解析】将数据分成若干份，每份单独处理后再合并，其思想为分治。

理想情况下，快速排序每次将数据划分为规模相近的两部分，并递归至不可再划分，因此其时间复杂度为 $O(n\log_2 n)$。在最坏情况下，每次划分都极不均匀，如一个类别中仅有一个元素，另一个类别中包含所有剩余元素。这时划分的复杂度为 $O(n)$，n 次操作的总复杂度为 $O(n_2)$。

【答案】(61)A；(62)D

8.6.5 堆排序

对于 n 个元素的关键字序列 $K_1, K_2, \cdots, K_n$，当且仅当所有关键字都满足下列性质时，称其为堆：

$$\begin{cases} K_i \leqslant K_{2i} \\ K_i \leqslant K_{2i+1} \end{cases}$$

或

$$\begin{cases} K_i \geqslant K_{2i} \\ K_i \geqslant K_{2i+1} \end{cases}, \quad i = 1, 2, \cdots, \left\lfloor \frac{n}{2} \right\rfloor$$

若堆顶为最小元素，则称为小根堆；若堆顶为最大元素，则称为大根堆。

堆排序的基本思想是：对一组待排序记录的关键字，首先把它们按堆的定义排成一个堆序列，从而输出堆顶的最小关键字(对于小根堆而言)，然后将剩余的关键字再调整成新堆，便得到次小的关键字，如此反复进行，直到全部关键字排成有序序列为止。

对于记录数较少的文件来说，堆排序的优越性并不明显，但对大量的记录来说堆排序是很有效的。堆排序的整个算法时间是由建立堆和不断调整堆这两部分时间代价构成的，堆排序算法的时间复杂度为 $O(n\log_2 n)$。此外，堆排序只需要一个记录大小的辅助空间。但是堆排序是一种不稳定的排序方法。

8.6.6 归并排序

所谓归并，是将两个或两个以上的有序文件合并成为一个新的有序文件。

归并排序是把一个有 n 个记录的无序文件看成由 n 个长度为 1 的有序子文件组成的文件，然后进行两两归并，如此重复，直至最后形成一个包含 n 个记录的有序文件为止。这种反复将两个有序文件归并成一个有序文件的排序方法称为两路归并排序。

8.6.7　基数排序

基数排序的思想是：设立 r 个队列，队列的编号分别为 $0,1,2,\cdots,r-1$。首先按最低有效位的值，把 n 个关键字分配到这 r 个队列中；然后从小到大将各队列中关键字再依次收集起来；接着按次低有效位的值把刚收集起来的关键字再分配到 r 个队列中。重复上述收集过程，直至最高有效位，这样得到了一个从小到大有序的关键字序列。为了减少记录移动的次数，队列可以采用链式存储分配，称为链队列。每个链队列设有两个指针，分别指向队头和队尾。

对于 n 个记录，执行一次分配和收集的时间为 $O(n+r)$，如果关键字有 d 位，则要执行 d 遍，所以总的运算时间为 $O(d(n+r))$。基数排序适用于链式分配的记录的排序，是一种稳定的排序方法。

8.6.8　内部排序方法的比较和选择

1. 内部排序方法的比较

内部排序方法的比较如表 8-2 所示。

表 8-2　内部排序方法的比较

排序方法	最好时间	平均时间	最坏时间	辅助空间	稳 定 性
直接插入	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
简单选择	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
冒泡排序	$O(n)$	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
希尔排序	—	$O(n^2)$	—	$O(1)$	不稳定
快速排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n^2)$	$O(n\log_2 n)$	不稳定
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(1)$	不稳定
归并排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n)$	稳定
基数排序	$O(d(n+rd))$	$O(d(n+rd))$	$O(d(n+rd))$	$O(rd)$	稳定

2. 内部排序方法的选择

选择排序方法时需要考虑的因素有：①待排序的记录个数 n；②记录本身的大小；③关键字的分布情况；④对排序稳定性的要求；⑤语言工具的条件、辅助空间的大小。

依据这些因素，可以得到以下几点结论。

- 若待排序的记录数目 n 较小时，可采用插入排序和选择排序。
- 若待排序记录按关键字基本有序，则宜采用直接插入排序或冒泡排序。
- 当 n 很大且关键字的位数较少时，采用链式基数排序较好。
- 若 n 较大，则应采用时间复杂度为 $O(n\log_2 n)$ 的排序方法：快速排序、堆排序或归并排序。

8.6.9 外部排序

常用的外部排序法是归并排序，这种方法一般分为两个阶段：在第一阶段，把文件中的记录分段读入内存，利用某种内部排序方法对这段记录进行排序并输出到外存的另一个文件中，在新文件中形成许多有序的记录段，称为归并段；在第二阶段，对第一阶段形成的归并段用某种归并方法进行一趟趟的归并，使文件的有序段逐渐加长，直到将整个文件归并为一个有序段时为止。

8.7 应试加油站

8.7.1 考频统计

通过对近几年考试真题的分析，“数据结构”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 8-3 所示。

表 8-3 历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：57、59～65	线性表、二叉查找树、模式匹配串、快速排序算法、哈夫曼编码	8
	下午：无		0
2014 年 5 月	上午：50、57～59、61	队列、线性表、二叉树的顺序存储结构、折半查找	5
	下午：无		0
2013 年 11 月	上午：57～63	线性表存储结构、循环队列、有向图拓扑序列、哈夫曼树哈希表、插入排序和快速排序算法	7
	下午：无		0
2013 年 5 月	上午：51～53、64～65	贪心算法和背包问题、分治算法	5
	下午：无		0
2012 年 11 月	上午：57～63	模式匹配、二叉树遍历、折半查找、拓扑排序、B-树、直接插入排序	7
	下午：无		0
2012 年 5 月	上午：21、57～61	数组、栈和队列操作、KMP 模式匹配算法、二叉树、图的存储结构、归并过程	6
	下午：无		0

纵观历年试卷，本章知识点是以选择题和综合分析题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量为 5～7 道选择题，所占分值为 5～7 分(占试卷总分值 75 分中的 7%～10%)。树、二叉树和图、排序算法、查找算法是本章考查的重点。

8.7.2　解题技巧

【典型题 8-1】采用顺序表和单链表存储长度为 n 的线性序列，根据序号查找元素，其时间复杂度分别为 __(51)__ 。(2013 年 5 月真题 51)

(51) A. $O(1)$、$O(1)$　　B. $O(1)$、$O(n)$　　C. $O(n)$、$O(1)$　　D. $O(n)$、$O(n)$

【解析】顺序表的存储位置是相邻连续的，是一种可以随机访问的数据结构，一个顺序表在使用前必须指定长度，一旦分配内存，则在使用中不可以动态地更改。它的优点是访问数据比较方便，可以随机地访问表中的任何一个数据。链表是通过指针来描述元素关系的一种数据结构，它可以是物理地址不连续的物理空间。不能随机访问链表元素，必须从表头开始，一步一步搜索元素。它的优点是：对于数组，可以动态地改变数据的长度，分配物理空间。因此两者的时间复杂度就显而易见了。

【答案】B

【典型题 8-2】设元素序列 a、b、c、d、e、f 经过初始为空的栈 S 后，得到出栈序列 c e d f b a，则栈 S 的最小容量为 __(52)__ 。(2013 年 5 月真题 52)

(52) A. 3　　B. 4　　C. 5　　D. 6

【解析】此题考查栈的用法，根据题中出栈的顺序，当元素 c 出栈后，栈中有元素 a、b，当元素 e 出栈之前，栈中有元素 a、b、d、e，此时栈中的元素达到最多。因此栈 S 的最小容量为 4。

【答案】B

【典型题 8-3】输出受限的双端队列是指元素可以从队列的两端输入、但只能从队列的一端输出，如图 8-10 所示。若有 e1、e2、e3、e4 依次进入输出受限的双端队列，则得不到输出队列 __(53)__ 。(2013 年 5 月真题 53)



图 8-10　双端队列

(53) A. e4、e3、e2、e1　　B. e4、e2、e1、e3

C. e4、e3、e1、e2　　D. e4、e2、e3、e1

【解析】此题考查队列的性质，队列为先进先出的线性结构，题中给出的受限的双端队列，两端都可以进，而一端可出，假设分 a 和 b 端，b 端可以进出，由 D 选项的输出序列，可以看出 e1、e2、e3 按顺序从 a 端进入，而 e4 从 b 端进入，当 e4 从 b 端出来之后，无法将后面的 e2 出队列，故 D 选项有误。

【答案】D

【典型题 8-4】以下关于线性表存储结构的叙述，正确的是 __(57)__ 。(2013 年 11 月真题 57)

(57) A. 线性表采用顺序存储结构时，访问表中任意一个指定序号元素的时间复杂度为常量级

B. 线性表采用顺序存储结构时，在表中任意位置插入新元素的运算时间复杂度为常量级

C. 线性表采用链式存储结构时，访问表中任意一个指定序号元素的时间复杂度为常量级

D. 线性表采用链式存储结构时，在表中任意位置插入新元素的运算时间复杂度为常量级

【解析】顺序存储结构可以随机存取，时间复杂度最低为常量级的，答案选 A。

【答案】A

【典型题 8-5】在字符串的模式匹配过程中，如果模式串的每个字符依次和主串中的一个连续的字符序列相等，则称为匹配成功。如果不能在主串中找到与模式串相同的子串，则称为匹配失败。在布鲁特-福斯模式匹配算法(朴素的或基本的模式匹配)中，若主串和模式串的长度分别为 n 和 m(且 n 远大于 m)，且恰好在主串末尾的 n 个字符处匹配成功，则在上述的模式匹配过程中，字符的比较次数最多为__(57)__。(2012 年 11 月真题 57)

(57) A. $n*m$　　B. $(n-m+1)*m$　　C. $(n-m-1)*m$　　D. $(n-m)*n$

【解析】 在最坏的情况下，每一趟不成功的匹配都是模式串的最后一个字符与主串中相应的字符不相等，则主串中新一趟的起始位置为 $i-m+2$。若从主串的第 i 个字符开始匹配时成功，则前 i 趟不成功的匹配中，每趟都比较了 m 次，总共比较了 $i*m$ 次，第 $i+1$ 趟的成功匹配也比较了 m 次。因此，在本题所述的匹配模式中，字符的比较次数最多为$(n-m+1)*m$ 次。

【答案】B

【典型题 8-6】对于一个长度大于 1 且不存在重复元素的序列，令其所有元素依次通过一个初始为空的队列后，再通过一个初始为空的栈。设队列和栈的容量都足够大，一个序列通过队列(栈)的含义是序列的每个元素都入队列(栈)且出队列(栈)一次且仅一次。对于该序列在上述队列和栈上的操作，正确的是__(57)__。(2012 年 5 月真题 57)

(57) A. 出队序列和出栈序列一定相同

B. 出队序列和出栈序列一定互为逆序

C. 入队序列和出队序列一定相同，入栈序列和出栈序列不一定相同

D. 入栈序列和出栈序列一定互为逆序，入队序列和出队序列不一定互为逆序

【解析】队列具有先进先出的特点，也就是说最先入队的元素最先出队，所以入队序列和出队序列一定相同。栈则具有先进后出的特点，如果所有元素进栈后再依次出栈，则入栈序列和出栈序列互为逆序，否则不一定。

【答案】C

【典型题 8-7】在字符串的 KMP 模式匹配算法中，需要求解模式串 p 的 next 函数值，其定义如下所示。若模式串 p 为“aaabaaa”，则其 next 函数值为__(58)__。(2012 年 5 月真题 58)

$$\text{next}[j]=\begin{cases}0, & j=1\\ \max\{k \mid 1<k<j, `p_1p_2\cdots p_{k-1}\text{'} = `p_{j-k+1}p_{j-k+2}\cdots p_{j-1}\text{'}\}\\ 1, & \text{其他情况}\end{cases}$$

(58) A. 0123123　　B. 0123210　　C. 0123432　　D. 0123456

【解析】j=1 时，next[1]=0。j=2 时，不存在 k，满足 $1<k<j$，则 next[2]=1。j=3 时，k 只能取 2，等式的左边为 p_1，等式的右边为 p_2，$p_1=p_2$=a，next[3]=2。j=4 时，k 可以取 2 和 3，k 取 2 的时候，左边为 p_1，右边为 p_3，$p_1=p_3$=a；k 取 3 时，左边为 p_1p_2，右边为 p_2p_3，$p_1p_2=p_2p_3$=aa；k 取较大值 3，因此 next[4]=3。j=5 时， k 可以取 2、3、2，k 取 2 的时候，左边为 p_1=a，右边为 p_4=b，左右两边不等；k 取 3 的时候，左边为 p_1p_2=aa，右边为 p_3p_4=ab，左右两边不等；k 取 4 的时候，左边为 $p_1p_2p_3$=aaa，右边为 $p_2p_3p_4$=aaba，左右两边不等，因此 next[5]=1。至此，可以判断正确的答案为 A。

【答案】A

【典型题 8-8】一个高度为 h 的满二叉树的结点总数为 2^h-1，从根结点开始，自上而下、同层次结点

从左至右，对结点按照顺序依次编号，即根结点编号为 1，其左、右孩子结点编号分为 2 和 3，再下一层从左到右的编号为 4、5、6、7，依次类推。那么，在一颗满二叉树中，对于编号为 m 和 n 的两个结点，若 n=2m+1，则 (64) 结点。(2013 年 5 月真题 64)

(64)A. m 是 n 的左孩子　　B. m 是 n 的右孩子　　C. n 是 m 的左孩子　　D. n 是 m 的右孩子

【解析】由于该二叉树为满二叉树，且根结点编号从 1 开始，由满二叉树的性质可知父结点 m 和右孩子之间的关系为 n=2m+1。

【答案】D

【典型题 8-9】以下关于哈夫曼树的叙述，正确的是 (60) 。(2013 年 11 月真题 60)

(60) A. 哈夫曼树一定是满二叉树，其每层结点数都达到最大值

B. 哈夫曼树一定是平衡二叉树，其每个结点左右子树的高度差为−1、0 或 1

C. 哈夫曼树中左孩子结点的权值小于父结点、右孩子结点的权值大于父结点

D. 哈夫曼树中叶子结点的权值越小则距离树根越远、叶子结点的权值越大则距离树根越近

【解析】哈夫曼树即最优二叉树，是一类带权路径长度的最短的树。树的带权路径为书中所有叶子结点的带权路径长度之和，记为

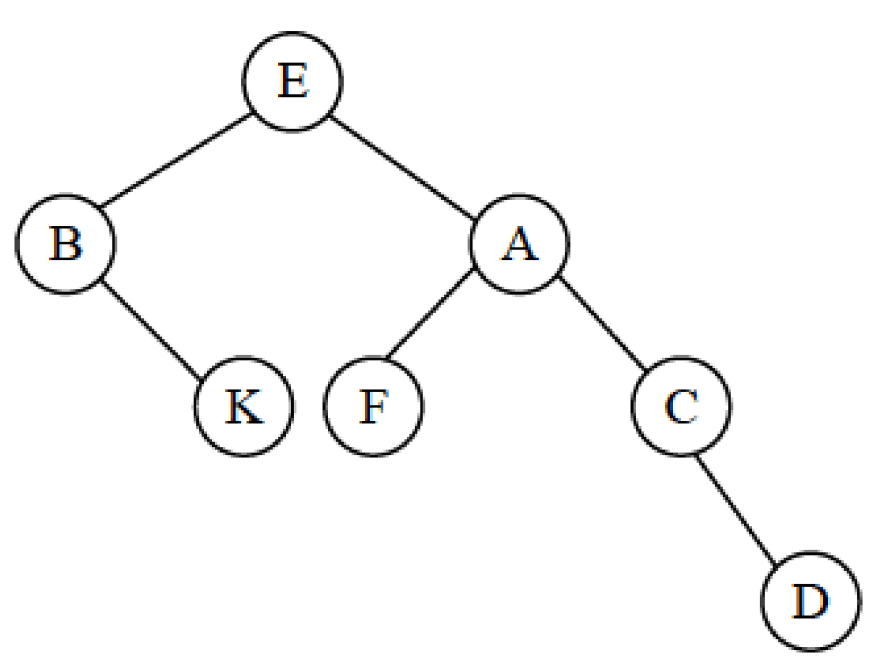
$$\text{WPL} = \sum_{k=1}^{n} w_k l_k$$

其中，n 为带权叶子结点的数目，w_k 为叶子结点的权值，l_k 为叶子结点到根的路径长度。哈夫曼树是指权值为 $w_1,w_2,\ldots,w_n$ 的 n 个叶子结点的二叉树中带权路径长度最小的二叉树。哈夫曼树与完全二叉树、平衡二叉树之间没有必然的联系。选项 A、B 中的说法是错误的。在哈夫曼树的构建中，由哈夫曼树的构造算法可知，哈夫曼树中权值最小的两个结点互为兄弟结点，根结点的权值为其左、右子树根结点的权值之和。

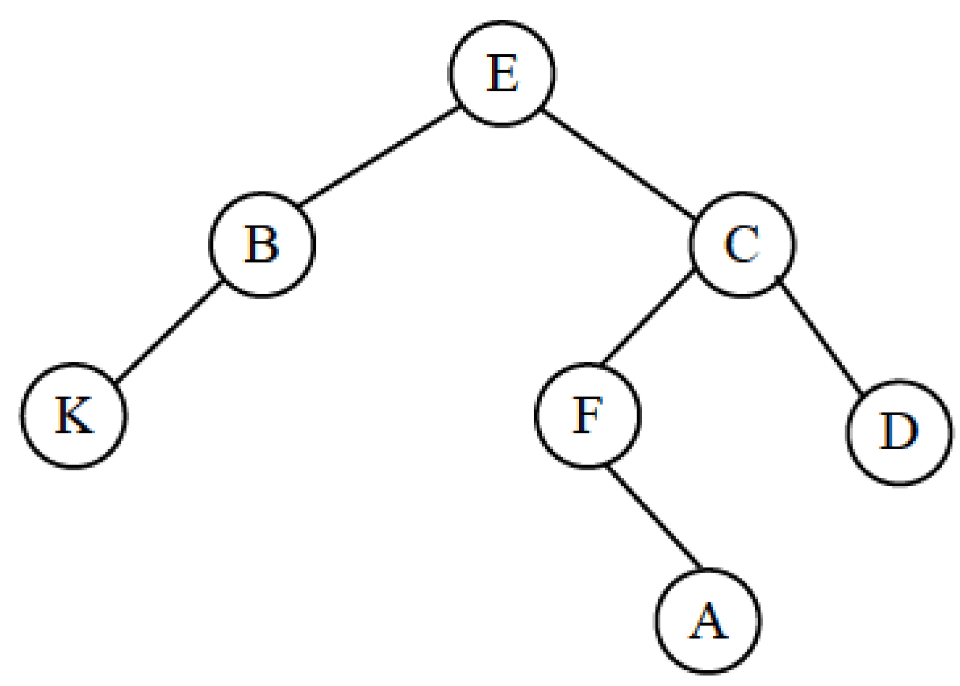
【答案】D

【典型题 8-10】若某二叉树的后序遍历序列为 KBFDCAE，中序遍历序列为 BKFEACD，则该二叉树为 (58) 。(2012 年 11 月真题 58)

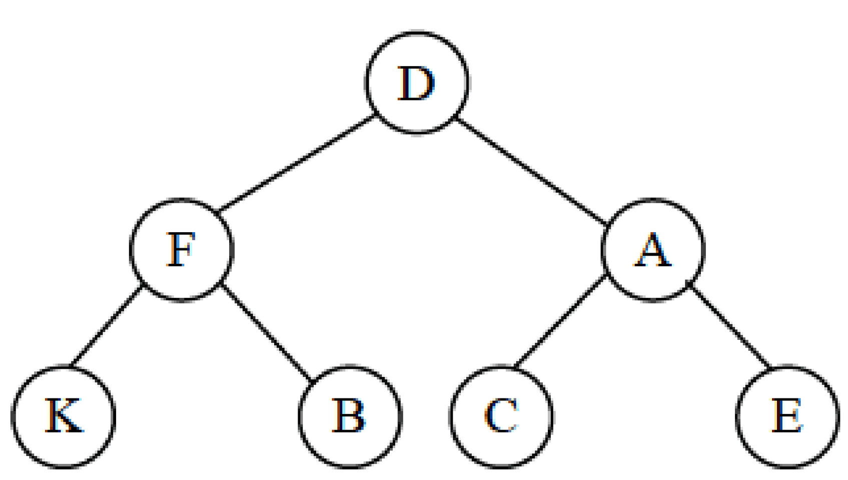
(58)　A.



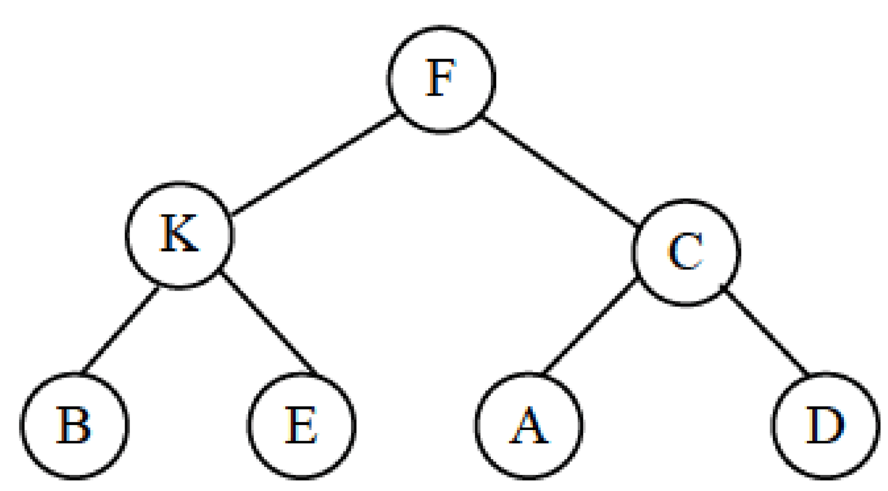
B.



C.



D.



【解析】 本题考查二叉树的遍历算法，根据中序遍历序列和另一种遍历序列的结果，可以确定该二叉树。后序遍历是按照左子树、右子树、根结点的顺序进行遍历，中序遍历是按照左子树、根结点、右子树的顺序进行遍历。E 为根结点，K 为 B 的右子树，因此应选 A 项描述的二叉树。

【答案】A

【典型题 8-11】若 n_2、n_1、n_0 分别表示一个二叉树中度为 2、度为 1 和叶子结点的数目(结点的度定义为结点的子树数目)，则对于任何一个非空的二叉树， (59) 。(2012 年 5 月真题 59)

(59) A. n_2 一定大于 n_1　　B. n_1 一定大于 n_0　　C. n_2 一定大于 n_0　　D. n_0 一定大于 n_2

【解析】由二叉树的性质可知，度为 0 的结点比度为 2 的结点数多 1，即 $n_0=n_2+1$，因此 n_0 一定大于 n_2。

【答案】D

【典型题 8-12】拓扑排序是将有向图中所有顶点排成一个线性序列的过程，并且该序列满足：若在 AOV 网中从顶点 V_i 到 V_j 有一条路径，则顶点 V_i 必然在顶点 V_j 之前。对于图 8-11 所示的有向图， (60) 是其拓扑序列。(2012 年 11 月真题 60)

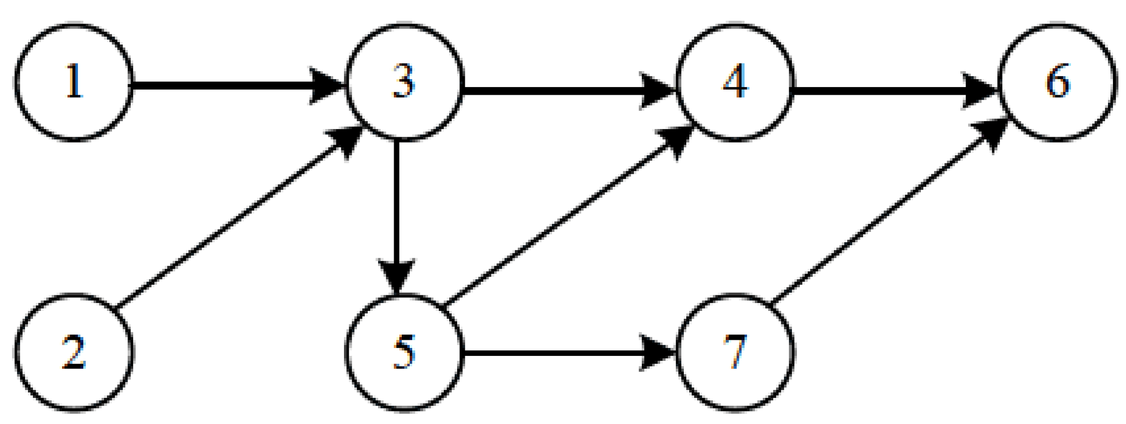


图 8-11　有向图

(60) A. 1234576　　B. 1235467　　C. 2135476　　D. 2134567

【解析】对 AOV 网进行拓扑排序的方法如下：

(1) 在 AOV 网中选择一个入度为 0(没有前驱)的顶点且输出它；

(2) 从网中删除该顶点及与该顶点有关的所有边；

(3) 重复上述两步，直至网中不存在入度为 0 的顶点为止。

本题中只有序列“2135476”是其拓扑序列。

【答案】C

【典型题 8-13】从存储空间的利用率角度来看，以下关于数据结构中图的存储的叙述，正确的是 (60) 。(2012 年 5 月真题 60)

(60) A. 有向图适合采用邻接矩阵存储，无向图适合采用邻接表存储

B. 无向图适合采用邻接矩阵存储，有向图适合采用邻接表存储

C. 完全图适合采用邻接矩阵存储

D. 完全图适合采用邻接表存储

【解析】邻接矩阵是用矩阵来指出顶点和顶点之间是否存在着关系。如果图有 n 个节点，则需要用 n^2 个元素来表示顶点间的关系。

邻接表是图的一种链式存储结构。在邻接表中，图中的每一个顶点都需要建立一个单链表，第 i 个单链表中的结点表示依附于顶点 V_i 的边。对于无向图，若无向图有 n 个顶点，e 条边，则它的邻接表需要 n 个头节点和 $2e$ 个表节点。对于有向图，若有 n 个顶点、e 条边，则它的邻接表需要 n 个头结点和 e 个表节点。当 $e<<n(n-1)/2$ 时，采用邻接表表示图比用矩阵节省空间。可见，完全图适合采用邻接矩阵存储。

【答案】C

【典型题 8-14】以下关于哈希(Hash，散列)查找的叙述中，正确的是__(65)__。(2013 年 5 月真题 65)

(65) A. 哈希函数应尽可能复杂些，以消除冲突

B. 构造哈希函数时应尽量使关键字的所有组成部分都能起作用

C. 进行哈希查找时，不再需要与查找表中的元素进行比较

D. 在哈希表中只能添加元素不能删除元素

【解析】哈希表中的元素是由哈希函数确定的。将数据元素的关键字 K 作为自变量，通过一定的函数关系(称为哈希函数)，计算出的值，即为该元素的存储地址。所以在构造哈希函数时应尽量使关键字的所有组成部分起作用。

【答案】B

【典型题 8-15】某哈希表(散列表)的长度为 n，设散列函数为 H(Key)=Key mod p，采用线性探测法解决冲突。以下关于 p 值的叙述中，正确的是__(61)__。(2012 年 11 月真题 61)

(61) A. p 的值一般为不大于 n 且最接近 n 的质数　　B. p 的值一般为大于 n 的任意整数

C. p 的值必须为小于 n 的合数　　D. p 的值必须等于 n

【解析】如果参数 p 是合数的话，那么 Key 相对于 p 的模得到的散列值会有很多是相同的，所以 p 一般取质数。如果 p 的值大于散列表的长度，散列函数得到的散列地址将和 Key 的范围相同大小，那么散列函数也就没有意义了，所以答案选 A。

【答案】A

【典型题 8-16】在 13 个元素构成的有序表 M[1..13]中进行折半查找(向下取整)，若找到的元素为 M[4]，则被比较的元素依次为__(59)__。(2012 年 11 月真题 59)

(59) A. M[7]、M[3]、M[5]、M[4]　　B. M[7]、M[5]、M[4]

C. M[7]、M[6]、M[4]　　D. M[7]、M[4]

【解析】 由于该有序表中共有 13 个元素，且元素下标为 1 至 13，即 low=1，high=13，用折半公式(low+high)/2(向下取整)，可以计算出首次被比较元素的下标是 7，即 M[7]。当与 M[7]比较完毕以后，发现不是要找的数据，所以继续查找。此时，low=1，high=6，用折半公式(low+high)/2 并向下取整，可以计算出首次被比较元素的下标是 3，即 M[3]。当与 M[3]比较完毕以后，发现不是要找的数据，所以继续查找。此时，low=4，high=6，用折半公式(low+high)/2 并向下取整，可以计算出首次被比较元素的下标是 5，即 M[5]。当与 M[5]比较完毕以后，发现不是要找的数据，所以继续查找，最终找到元素 M[4]。

【答案】A

【典型题 8-17】图 8-12 所示为一棵 N 阶 B⁻树，N 最有可能的值为__(61)__。(2012 年 11 月真题 61)

(61) A. 1　　B. 2　　C. 3　　D. 4

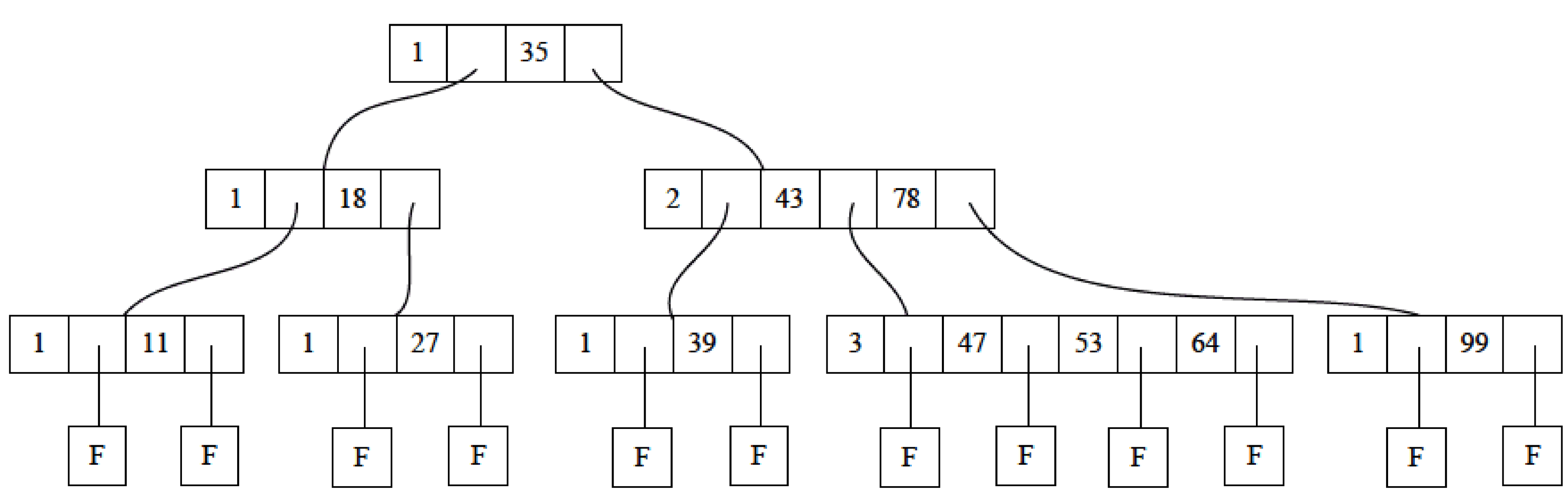


图 8-12　N 阶 B 树

【解析】一棵 N 阶 B^{-} 树为满足以下特性的 N 叉树：

(1) 树中每个节点至多有 N 棵子树；

(2) 若根节点不是叶子节点，则至少有两棵子树；

(3) 除根之外的所有非终端节点至少有 $\left\lceil \frac{N}{2} \right\rceil$ 棵子树；

(4) 所有的非终端节点中包含下列数据信息$(n,A_0,K_1,A_1,K_2,A_2,\cdots,K_n,A_n)$。其中，$K_i(i=1,2,\cdots,n)$为关键字(如 3,47,53,63)，且 $K_i<K_{i+1}$，$A_i(i=0,1,2,\cdots,n)$为指向子树根节点的指针，n 为节点中关键字的个数，且$\left\lceil \frac{N}{2} \right\rceil-1 \leqslant n \leqslant N-1$。

(5) 所有的叶子节点都出现在同一层次上，并且不带信息。

由图8-10可知，N最有可能的值为4。

【答案】D

【典型题 8-18】对 n 个基本有序的整数进行排序，若采用插入排序算法，则时间和空间复杂度分别为__(62)__；若采用快速排序算法，则时间和空间复杂度分别为__(63)__。(2013 年 11 月真题 62～63)

(62) A. $O(n^2)$和$O(n)$　　B. $O(n)$和$O(n)$

C. $O(n^2)$和$O(1)$　　D. $O(n)$和$O(1)$

(63)A. $O(n^2)$和$O(n)$　　B. $O(n\lg n)$和$O(n)$

C. $O(n^2)$和$O(1)$　　D. $O(n\lg n)$和$O(1)$

【解析】插入排序的基本操作就是将一个数据插入到已经排好序的有序数据中，从而得到一个新的、个数加一的有序数据。若数据基本有序，对插入排序算法而言，可以在近似线性时间内完成排序，时间复杂度为 $O(n)$。对于快速排序而言，数据基本有序可以看成是最坏的情况，时间复杂度为 $O(n^2)$。对于两个排序算法，在排序时都需要一个额外的存储空间，因此空间复杂度均为 $O(1)$。

【答案】(62)D；(63)C

【典型题 8-19】将数组{1,1,2,4,7,5}从小到大排序，若采用__(62)__排序算法，则元素之间需要进行的比较次数最少，共需要进行__(63)__次元素之间的比较。(2012 年 11 月真题 62～63)

(62) A. 直接插入　　B. 归并　　C. 堆　　D. 快速

(63) A. 5　　B. 6　　C. 7　　D. 8

【解析】 直接插入排序算法的基本思想是将待排序数组分为两个部分：已排好序部分和未排序部分。其主要步骤为：开始时，第一个元素在已排好序部分中，其余部分在未排序部分。然后依次从未排序部分中取出第一个元素，从后向前与排好序部分的元素进行比较并将其插入到已排好序部分的正确位置，直到所有元素排好序。当序列基本有序时，直接插入排序过程中元素比较的次数较少；当序列为逆序时，元素的比较次数最多。使用直接插入排序算法，数组{1,1,2,4,7,5}需要比较 6 次，依次为 1 与 1 比较、2 与 1 比较、4 与 2 比较、7 与 4 比较、5 与 7 比较、5 与 4 比较。

并归排序的基本思想是将待排序数组划分为子问题，对子问题求解，然后合并解。其主要步骤为：将数组分为两个相同规模的子数组，分别包含前 $n/2$ 个元素和后 $n/2$ 个元素；递归地排序这两个子数组；合并排好序的两个子数组，依次比较两个排好序的子数组的元素，得到整个数组的排好序的序列。使用直接插入排序算法，数组{1,1,2,4,7,5}需要比较 8 次。

【答案】(62)A；(63)B

【典型题 8-20】递增序列 $A(a_1,a_2,\cdots,a_n)$和 $B(b_1,b_2,\cdots,b_n)$的元素互不相同，若需将它们合并为一个长度为 $2n$ 的递增序列，则当最终的排列结果为__(61)__时，归并过程中元素的比较次数最多。(2012 年 5 月真题 61)

(61) A. $a_1,a_2,\cdots,a_n,b_1,b_2,\ \cdots,b_n$

B. $b_1,b_2,\ \cdots,b_n,a_1,a_2,\cdots,a_n$

C. $a_1,\ b_1,a_2,\ b_2,\cdots,a_i,b_i,\cdots,\ a_n,b_n$

D. $a_1,a_2,\cdots,a_{i/2},b_1,b_2,\ \cdots,b_{i/2},\ a_{i/2+1},\ a_{i/2+2},\cdots a_n,b_{i/2+1},\ b_{i/2+2},\ \cdots,b_n$

【解析】归并排序是将两个排好序的序列合并成一个有序的序列。由选项 A 给出的结果可知，递增序列 B 的每一个元素都比 A 中的元素要大，也就是说 $a_i(1\leqslant i\leqslant n)$比 b_1 小， 在排序的过程中，只需要将 a_i 与 b_1 进行比较，共比较了 n 次。由选项 B 给出的结果可知，递增序列 B 的每一个元素都比 A 中的元素要小，在排序的过程中，只需要将 $b_i(1\leqslant i\leqslant n)$与 a_1 进行比较，共比较了 n 次。由选项 C 给出的结果可知，$a_i<b_i<a_{i+1}$，在排序的过程中，将 a_1 与 b_1 进行比较，a_1 小，然后将 a_2 与 b_1 进行比较，a_2 大，则已排好的部分为 a_1b_1，共比较了 2 次；然后将 a_2 与 b_2 进行比较，a_2 小，再将 a_3 与 b_2 进行比较，a_3 大，则已排好的部分为 $a_1b_1a_2b_2$，共比较了 4 次；以此类推，完全排好时共比较了 $2(n-1)+1=2n-1$ 次。由选项 D 给出的结果可知，递增序列 B 的前 $i/2$ 个元素都比 A 中的前 $i/2$ 个元素要大，但比 A 中的后 $i/2$ 个元素要小，B 的后 $i/2$ 个元素都比 A 中的后 $i/2$ 个，因此在排序的时候，a 的前 $i/2$ 个元素只需与 b_1 进行比较，当比较到 $a_{i/2+1}$ 时，$a_{i/2+1}$ 比 b_1 大，则已排好的部分为 $a_1,a_2,\cdots,a_{i/2}$，共比较了 $i/2+1$ 次；然后将 $b_2,b_3,\cdots,b_{i/2}$ 与 $a_{i/2+1}$ 进行比较，都比与 $a_{i/2+1}$ 小，当比较到到 $b_{i/2+1}$ 时，$b_{i/2+1}$ 比 $a_{i/2+1}$ 大，则已排好的部分为 $a_1,a_2,\cdots,a_{i/2},b_1,b_2,\ \cdots,b_{i/2}$，共比较了 $i+1$ 次；然后将 $a_{i/2+2}\ \cdots,a_n$ 分别与 $b_{i/2+1}$ 进行比较，共比较了 $n-i/2-2$ 次，完全排好时共比较了 $i+1+\ n-i/2-2=n+i/2-1$ 次。

可见，对于选项 A、B、C、D 中的结果，需比较的次数分别为 n，n，$2n-1$，其中 $2n-1$ 最大，即选项 C 中的结果要比较的次数最多。

【答案】C

8.8　过 关 习 题

1. 对于线性表(由 n 个同类元素构成的线性序列)，采用单向循环链表存储的特点之一是______。

   A. 从表中任意结点出发都能遍历整个链表

   B. 对表中的任意结点可以进行随机访问

   C. 对于表中的任意一个结点，访问其直接前驱和直接后继结点所用时间相同

   D. 第一个结点必须是头结点

2. 设循环队列 Q 的定义中有 rear 和 len 两个域变量，其中 rear 表示队尾元素的指针，len 表示队列的长度，如图 8-13 所示(队列长度为 3，队头元素为 e)。设队列的存储空间容量为 M，则队头元素的指针为______。

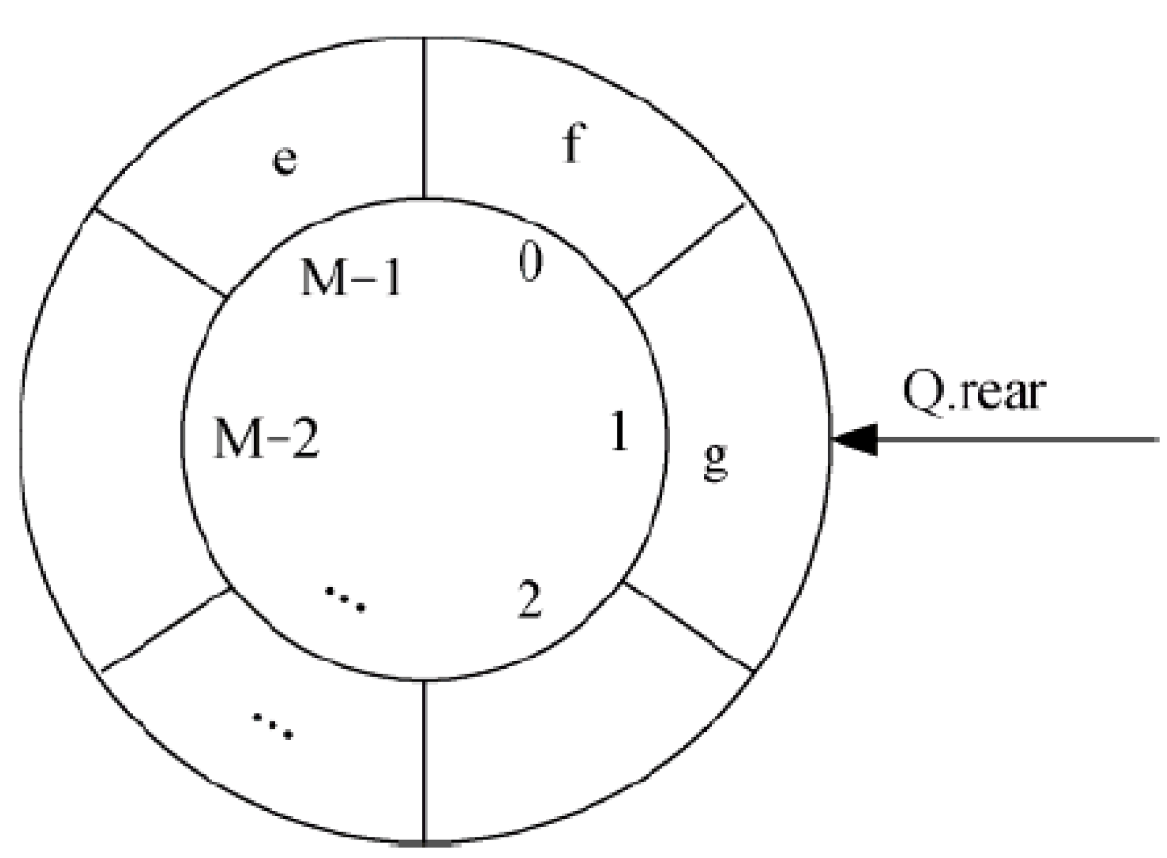


图 8-13　循环队列 Q

A. (Q.rear+Q.len−1)　　B. (Q.rear+Q.len−1+M)%M

C. (Q.rear−Q.len+1)　　D. (Q.rear−Q.len+1+M)%M

3. 在 KMP 模式匹配算法中，需要求解模式串 p 的 next 函数值，其定义如下(其中，j 为模式串字符的序号)。对于模式串“abaabaca”，其 next 函数值序列为______。

$$\text{next}[j]=\begin{cases}0, & j=1\\ \max\{k\mid 1<k<j, \text{'}p_1p_2\cdots p_{k-1}\text{'}=\text{'}p_{j-k+1}p_{j-k+2}\cdots p_{j-1}\text{'}\} \\ 1, & \text{其他情况}\end{cases}$$

A. 01111111　　B. 01122341　　C. 01234567　　D. 01122334

4. 若二维数组 arr[1..M, 1..N]的首地址为 base，数组元素按列存储且每个元素占用 K 个存储单元，则元素 arr[i, j]在该数组空间的地址为______。

A. base+((i−1)*M+j−1)*K　　B. base+((i−1)*N+j−1)*K

C. base+((j−1)*M+i−1)*K　　D. base+((j−1)*N+i−1)*K

5. 设下三角矩阵(上三角部分的元素值都为 0)A[0..n, 0..n]如下所示，将该三角矩阵的所有非零元素(即行下标不小于列下标的元素)按行优先压缩存储在容量足够大的数组 M[1..m]中，则元素 $A[i,j](0\leqslant i\leqslant n, j\leqslant i)$存储在数组 M 的______中。

$$\begin{bmatrix} A_{00} & & & & & \\ A_{10} & A_{11} & & & 0 & \\ \bullet & & \bullet & & & \\ \bullet & & & \bullet & & \\ \bullet & & & & \bullet & \\ A_{70} & A_{71} & A_{72} & & \cdots & A_{77} & \\ A_{80} & A_{81} & A_{82} & A_{83} & & \cdots & A_{88} \end{bmatrix}$$

A. $M\left[\dfrac{i(i+1)}{2}+j+1\right]$　　B. $M\left[\dfrac{i(i+1)}{2}+j\right]$

C. $M\left[\dfrac{i(i-1)}{2}+j\right]$　　D. $M\left[\dfrac{i(i-1)}{2}+j+1\right]$

6. 已知一棵度为 3 的树(一个结点的度是指其子树的数目，树的度是指该树中所有结点的度的最大值)中有 5 个度为 1 的结点，4 个度为 2 的结点，2 个度为 3 的结点，那么，该树中的叶子结点数目为______。

A. 10　　B. 9　　C. 8　　D. 7

7. 一棵满二叉树，其每一层结点个数都达到最大值，对其中的结点从 1 开始顺序编号，即根结点编号为 1，其左、右孩子结点编号分别为 2 和 3，再下一层从左到右的编号为 4、5、6、7，以此类推，每一层都从左到右依次编号，直到最后的叶子结点层为止，则用______可判定编号为 m 和 n 的两个结点是否在同一层。

A. $\log_2 m=\log_2 n$　　B. $\lfloor\log_2 m\rfloor=\lfloor\log_2 n\rfloor$

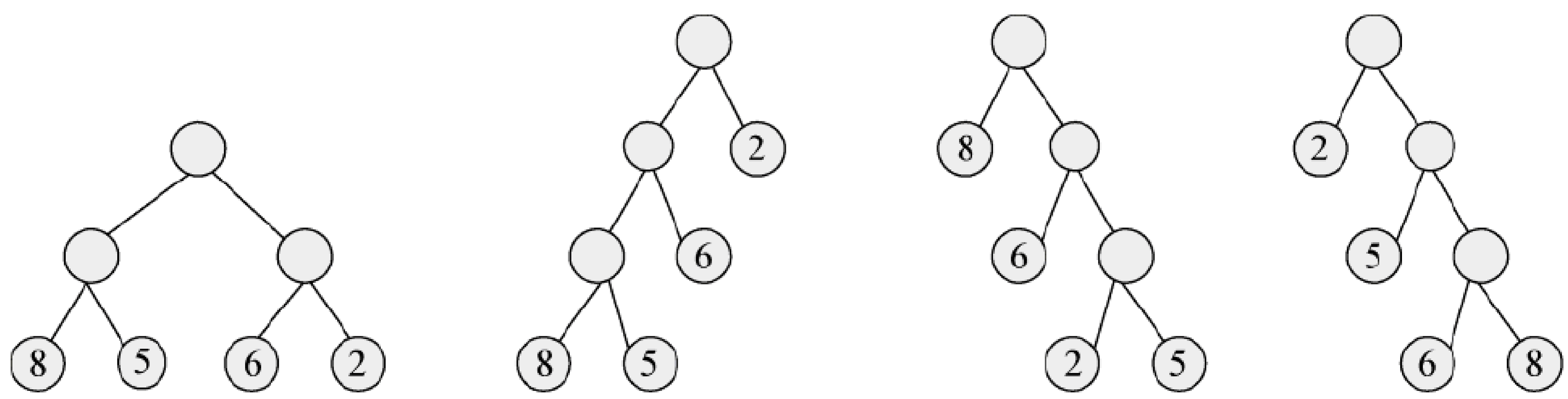
C. $\lfloor\log_2 m\rfloor+1=\lfloor\log_2 n\rfloor$　　D. $\lfloor\log_2 m\rfloor=\lfloor\log_2 n\rfloor+1$

8. 在______中，任意一个结点的左、右子树的高度之差的绝对值不超过 1。

A. 完全二叉树　　B. 二叉排序树　　C. 线索二叉树　　D. 最优二叉树

9. ______是由权值集合{8,5,6,2}构造的哈夫曼树(最优二叉树)。

A.　　　　B.　　　　C.　　　　D.



10. 下面关于哈夫曼树的叙述中，正确的是______。

A. 哈夫曼树一定是完全二叉树

B. 哈夫曼树一定是平衡二叉树

C. 哈夫曼树中权值最小的两个结点互为兄弟结点

D. 哈夫曼树中左孩子结点小于父结点，右孩子结点大于父结点

11. 无向图中一个顶点的度是指图中与该顶点相邻接的顶点数。若无向图 G 中的顶点数为 n，边数为 e，则所有顶点的度数之和为______。

A. $n\times e$　　B. $n+e$　　C. $2n$　　D. $2e$

12. 设一个包含 N 个顶点、E 条边的简单无向图采用邻接矩阵存储结构(矩阵元素 $A[i][j]$等于 1/0 分别表示顶点 i 与顶点 j 之间有/无边)，则该矩阵中的非零元素数据为______。

A. N　　B. E　　C. $2E$　　D. $N+E$

13. ______是图 8-14 的合法拓扑序列。

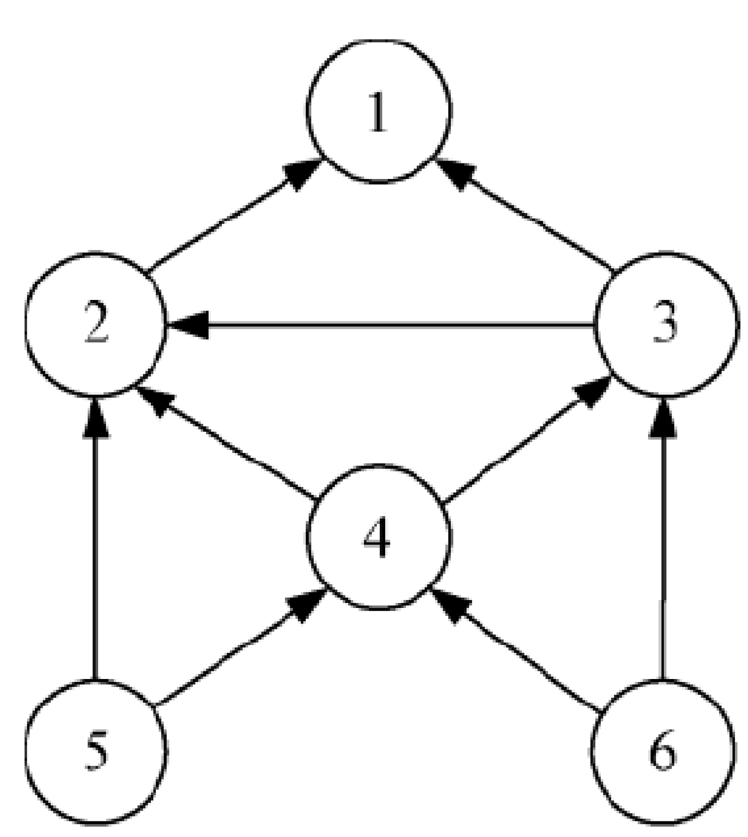


图 8-14　有向图

A. 6 5 4 3 2 1　　B. 1 2 3 4 5 6　　C. 5 6 3 4 2 1　　D. 5 6 4 2 1 3

14. 对 n 个元素的有序表 $A[1..n]$进行顺序查找，其成功查找的平均查找长度(即在查找表中找到指定关键码的元素时，所进行比较的表中元素个数的期望值)为______。

A. n　　B. $(n+1)/2$　　C. $\log_2 n$　　D. n^2

15. 某一维数组中依次存放了数据元素 15,23,38,47,55,62,88,95,102,123，采用折半(二分)法查找元素 95 时，依次与______进行了比较。

A. 62,88,95　　B. 62,95　　C. 55,88,95　　D. 55,95

16. 对于关键字序列(26, 25, 72, 38, 8, 18, 59)，采用散列函数 $H(\text{Key})=\text{Key mod } 13$ 构造散列表(哈希表)。

若采用线性探测的开放定址法解决冲突(顺序地探查可用存储单元)，则关键字 59 所在散列表中的地址为______。

A. 6　　B. 7　　C. 8　　D. 9

17. 用插入排序和归并排序算法对数组<3,1,4,1,5,9,6,5>从大到小进行排序，则分别需要进行______次数组元素之间的比较。

A. 12, 14　　B. 10, 14　　C. 12,16　　D. 10, 16

第 9 章

算法与设计

本章要点

- 算法的时间复杂度和空间复杂度的计算
- 递归式
- 分治算法
- 动态规划算法
- 贪心法
- 回溯法

9.1 算法设计与分析基础

考核说明：本节主要介绍算法的概念、时间复杂度、渐进符号、递归式、算法设计与分析等内容。其中，时间复杂度和空间复杂度、递归式是考试的重点和难点，要能理解和掌握。

9.1.1 算法

1. 算法的概念

算法是对特定问题求解步骤的一种描述，它是指令的有限序列，其中每一条指令表示一个或多个操作。一个算法具有下列五个重要特性。

- 有穷性：一个算法必须总是在执行有穷步之后结束，且每一步都可在有穷时间内完成。
- 确定性：算法中的每一条指令必须有确切的含义，读者理解时不会产生二义性，并且在任何条件下，算法只有唯一的一条执行路径，即对于相同的输入只能得出相同的输出。
- 可行性：一个算法是可行的，即算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现的。
- 输入：一个算法有零个或多个输入，这些输入取自某个特定的对象的集合。
- 输出：一个算法有一个或多个输出，这些输出是同输入有着某些特定关系的量。

2. 时间复杂性

算法的时间复杂度分析主要是分析算法的运行时间，即算法所执行的基本操作数。

根据不同的输入，将算法的时间复杂度分析分为三种情况。

(1) 最佳情况是算法执行时间最少的输入。一般情况下，不进行算法在最佳情况下的时间复杂度分析。

(2) 最坏情况是算法执行时间最多的输入。一般会进行算法在最坏时间复杂度的分析，因为最坏情况是在任何输入下运行时间的一个上限，它给我们提供了一个保障，情况不会比这更糟糕。

(3) 平均情况是算法的平均运行时间。一般来说，这种情况很难分析。平均情况分析可以按如下三个步骤进行。

① 将所有的输入按其执行时间分类。

② 确定每类输入发生的概率。

③ 确定每类输入的执行时间。

下式给出了一般算法在平均情况下的复杂度分析。

$$T(n)=\sum_{i=1}^{m}p_i\times t_i$$

式中：p_i为第 i 类输入发生的概率；t_i为第 i 类输入的执行时间，输入分为 m 类。

3. 渐进符号

渐进符号有如下几种。

(1)　O 记号。给出一个函数的渐进上界。

(2)　Ω 记号。给出一个函数的渐进下界。

(3)　Θ 记号。给出一个函数的渐进上界和下界，即渐进确界。

4. 递归式

从算法的结构上看，算法可以分为非递归形式和递归形式。非递归算法的时间复杂度分析较简单，本节主要讨论递归算法的时间复杂度分析方法。

(1)　展开法。将递归式中等式右边的项根据递归式进行替换，称为展开。展开后的项被再次展开，如此下去，直到得到一个求和表达式，得到结果。

(2)　代换法。这一名称来源于当归纳假设用较小值时，用所猜测的值代替函数的解。用代换法解递归式时，需要两个步骤：猜测解的形式；用数学归纳法找出使解真正有效的常数。

(3)　递归树法。递归树法弥补了代换法猜测困难的缺点，它适于提供“好”的猜测，然后用代换法证明。在递归树中，每一个结点都代表递归函数调用集合中每一个子问题的代价。将树中每一层内的代价相加得到一个每层代价的集合，再将每层的代价相加得到递归式所有层次的总代价。当用递归式表示分治算法的时间复杂度时，递归树的方法尤其有用。

(4)　主方法。也称为主定理，给出求解如下形式的递归式的快速方法：

$$T(n)=aT(n/b)+f(n)$$

式中：$a\geqslant 1$ 和 $b>1$ 是常数；$f(n)$是一个渐进的正函数。

5. 算法的表示

常用的表示算法的方法有以下几种。

- 自然语言。
- 流程图。
- 程序设计语言。
- 伪代码。

真题链接

【例 9-1】 以下关于渐近符号的表示中，不正确的是__(62)__。(2012 年 5 月真题 62)

(62) A. $n^2=\Theta(n^2)$　　B. $n^2=O(n^2)$　　C. $n^2=O(n)$　　D. $n^2=O(n^3)$

【解析】 如果存在正常数 c 和 n_0，使得当 $n\geqslant n_0$ 时，$T(n)\leqslant cf(n)$，则记为 $T(n)=O(f(n))$。T 和 f 的关系可以理解为 $f(n)$为 $T(n)$的一个上界，也可以理解为 T 至多增长得和 f 一样快。

如果存在正常数 c_1，c_2 和 n_0，使得当 $n\geqslant n_0$ 时，$c_1f(n)\leqslant T(n)\leqslant c_2f(n)$，则记为 $T(n)=\Theta(f(n))$。T 与 f 有着相同的阶数，或者两者最终与相同的阶数增长。

对于选项 A，$T(n)=f(n)=n^2$，只要 $c_2\geqslant c_1\geqslant 1$，$n_0>0$，就有 $c_1f(n)\leqslant T(n)\leqslant c_2f(n)$，因此有 $T(n)=\Theta(f(n))$，即 $n^2=\Theta(n^2)$。

对于选项 B，$T(n)=f(n)=n^2$，只要 $c\geqslant1$，$n_0>0$，就有 $T(n)\leqslant cf(n)$，因此有 $T(n)=O(f(n))$，即 $n^2=O(n^2)$。

对于选项 D，$T(n)=n^2$，$f(n)=n^3$，只要 $c\geqslant1$，$n_0>1$，就有 $T(n)\leqslant cf(n)$，因此有 $T(n)=O(f(n))$，即 $n^2=O(n^3)$。

对于选项 C，当 $n>1$ 时，n^2 的增长比 n 快，因此 $n^2=O(n)$的关系不成立。

【答案】C

9.1.2 算法设计与分析

1. 算法设计

通常求解一个问题可能会有多种算法可供选择，选择的主要标准首先是算法的正确性和可靠性、简单性和易理解性；其次是算法所需要的存储空间少和执行更快等。

算法设计是一件非常困难的工作，通常设计一个“好”的算法应考虑达到以下目标：正确性、可读性、健壮性、效率与低存储量需求。

经常采用的算法设计技术主要有迭代法、穷举搜索法、递推法、贪心法、回溯法、分治法和动态规划法等。

2. 算法分析

算法分析是指对一个算法所需要的资源进行估算，这些资源包括内存、通信带宽、计算机硬件和时间等，所需要的资源越多，该算法的复杂性就越高。不言而喻，对于任何给定的问题，设计出复杂性尽可能低的算法是设计算法时追求的重要目标。另外，当给定问题有很多种算法时，选择其中复杂性最低者，是选用算法时应遵循的重要准则。

在计算机资源中，最重要的是时间和空间(存储器)资源，因此复杂性分析主要包括时间复杂性和空间复杂性。

真题链接

【例 9-2】某个算法的时间复杂度递归式 $T(n)=T(n-1)+n$，其中 n 为问题的规模，则该算法的渐进时间复杂度为 (62) ，若问题的规模增加了 16 倍，则运行时间增加 (63) 倍。(2014 年 5 月真题 62～63)

(62) A. $O(n)$　　B. $O(n\lg n)$　　C. $O(n^2)$　　D. $O(n^2\lg n)$

(63) A. 16　　B. 64　　C. 256　　D. 1024

【解析】对于递归式，假设 $T(1)=1$，则

$$
\begin{aligned}
T(n)&=T(n-1)+n\\
&=T(n-2)+n-1+n\\
&=T(n-3)+n-2+n-1+n\\
&=\dots\\
&=1+2+\dots+n-1+n\\
&=n(n+1)/2
\end{aligned}
$$

可见，时间复杂度为 $O(n^2)$。若问题的规模增加了 16 倍，则运行时间增加了 $16^2=256$ 倍。

【答案】(62)C；(63)C

9.2　分　治　法

考核说明：本节主要介绍分治算法的基本思想和应用，经常会考到，要能读懂 Hanoi 塔问题求解的 C 代码实现。

9.2.1　递归的概念

递归是指子程序(或函数)直接调用自己或通过一系列调用语句间接调用自己，是一种描述问题和解决问题的常用方法。

递归有两个基本要素：边界条件，即确定递归到何时终止，也称为递归出口；递归模式，即大问题是如何分解为小问题的，也称为递归体。

9.2.2　分治法的基本思想

分治法的设计思想是将一个难以直接解决的大问题分解成一些规模较小的相同问题，以便各个击破，分而治之。如果规模为 n 的问题可分解成 k 个子问题，$1<k\leqslant n$，这些子问题互相独立且与原问题相同。分治法产生的子问题往往是原问题的较小模式，这就为递归技术提供了方便。

一般来说，分治算法在每一层递归上都有三个步骤。

(1) 分解。将原问题分解成一系列子问题。

(2) 求解。递归地求解各子问题。若子问题足够小，则直接求解。

(3) 合并。将子问题的解合并成原问题的解。

9.2.3　典型应用：Hanoi 塔问题

Hanoi 塔问题描述如下：有 n 个盘子在 a 处，盘子从大到小，最上面的盘子最小。现在要把这 n 个盘子从 a 处搬到 c 处，可以在 b 处暂存，但每次只能移动一个盘子，任何时候都不能出现大的盘子压在小的盘子上面的情况。

当只有一个盘子时，直接从 a 移到 c 即可；如果已知 $n-1$ 个盘子的移动方案，那么 n 个盘子的移动方案如下：先把前 $n-1$ 个盘子从 a 借助 c 移动到 b 处，再把第 n 个盘子从 a 处直接移动到 c 处，然后再将 b 处的 $n-1$ 个盘子从 b 处借助 a 处移动到 c 处。至此就完成了全部盘子的移动。具体 C 代码实现如下：

```
void Hanoi(int n, char a, char b, char c)//将 n 个盘子从 a 通过 b 移动到 c
{
    if(n > 1){
        Hanoi(n-1, a, c, b); //先将前 n-1 个盘子从 a 处通过 c 移动到 b 处
        move(n, a, c);       //将第 n 个盘子从 a 处直接移动到 c 处
        Hanoi(n-1, b, a, c); //再将前 n-1 个盘子从 b 处通过 a 移动到 c 处
```

```
    }else{
        move(n, a, c);        //只有一个盘子时，直接从 a 移动到 c。递归出口
    }
}
```

真题链接

【例 9-3】给定 n 个整数构成的数组 $A=\{a_1, a_2, \cdots, a_n\}$和整数 x，判断 A 中是否存在两个元素 a_i 和 a_j，使的 $a_i+a_j=x$。为了求解问题，首先用归并排序算法对数组 A 进行从大到小排序；然后判断是否存在 $a_i+a_j=x$，具体的方法如下列伪代码所示。则求解该问题时排序算法应用了 (62) 算法设计策略，整个算法的时间复杂度为 (63) 。(2013 年 5 月真题 62～63)

```
…
i=1;j=n
While i<j
  If ai+aj=x return true
   Else if ai+aj>x
        J--;
   Else
        I++;
  Return false;
```

(62) A. 分治　B. 贪心　C. 动态规划　D. 回溯

(63) A. $O(n)$　B. $O(n\lg n)$　C. $O(n^2)$　D. $O(n\lg^2 n)$

【解析】分治算法的基本思想是将一个规模为 N 的问题分解为 K 个规模较小的子问题，这些子问题相互独立且与原问题性质相同。求出子问题的解，就可得到原问题的解。分支算法的时间复杂度为 $O(n\lg n)$。

【答案】(62)A；(63)B

【例 9-4】(2014 年 5 月试题四)

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

采用归并排序对 n 个元素进行递增排序时，首先将 n 个元素的数组分成各含 $n/2$ 个元素的两个子数组，然后用归并排序对两个子数组进行递归排序，最后合并两个已经排序的子数组得到排序结果。

下面的 C 代码是对上述归并算法的实现，其中的常量和变量说明如下：

arr：待排序数组。

p,q,r：一个子数组的位置从 p 到 q，另一个子数组的位置从 q+1 到 r。

begin,end：待排序数组的起止位置。

left,right：临时存放待合并的两个子数组。

n1,n2：两个子数组的长度。

i,j,k：循环变量。

mid：临时变量。

【C 代码】

```
#inciude<stdio.h>
#inciude<stdlib.h>
```

```
Define MAX 65536
void merge(int arr([ ],int p,int q,int r) {
int *left, *right;
int  n1,n2,I,j,k;
n1= q - p +1;
n2= r -q;
If(left = (int*)malloc((n1+1)*sizeof(int))) = NULL) {
Perror( "malloc error" );
exit(1);
}
If((right = (int*)malloc((n2 +1)*sizeof(int))) = NULL) {
Perror("malloc error");
exit(1);
}
for(i=0;i<n1;i++){
left[ i ] = arr [p+i];
}
left[ i ]=MAX;
for(i=0; i<n2; i++){
right[i]= arr[q + i+ 1]
}
right[ i ]=MAX;
i=0;j=0;
For(k=p;(1);k++){
If(left[i]> right[j] {
 (2) 
j++;
}else{
arr[k1] =left[i];
i++;
}
}
}
Void merge Sort(int arr( ),int begin,int end){
int mid;
if(  (3)  ){
mid = (begin + end) / 2;
merge Sort(arr, begin, mid);
 (4)  ;
Merge(arr, begin, mid, end);
}
}
```

【问题 1】

根据以上说明和 C 代码，填充(1)～(4)，

【问题 2】

根据题干说明和以上 C 代码，算法采用了 (5) 算法设计策略，分析时间复杂度时，列出其递归式为 (6) ，解出渐近时间复杂度为 (7) (用 O 符号表示)，空间复杂度为 (8) (用 O 符号表示)。

【问题 3】

两个长度分别为 $n1$ 和 $n2$ 的已经排好序的子数组进行归并，根据上述 C 代码，则元素之间比较次数为 (9) 。

【解析】

【问题 1】

首先，函数 void merge(int arr[],int p,int q,int r)的意思是：对子数组 arr[p..q]和子数组 arr[q+L..r]进行合并，因此第一空为 k<=q；由于是采用归并排序对 n 个元素进行递增排序，所以第二空是将 left[i]和 right[j]中的小者存放到 arr[k]中去，即 arr[k] =right[j]；当数组长度为 1 时，停止递归，因为此时该数组有序，则第三空为 begin<end，即数组至少有两个元素才进行递归，合并了 begin 到 mid 之间的元素，继续合并 mid+l 到 end 之间的元素，则第四空为 mergeSort(arr,mid+l,end)。

【问题 2】

归并算法实际上就是将数组一直往下分割，直到分割到由一个元素组成的 n 个子数组，再往上两两归并。

将数组进行分割需要 $\log_2 n$ 步，因为每次都是将数组分割成两半($2x=n$，$x = \log_2 n$)。

合并 n 个元素，需要进行 n 步，也就是 $O(n)$，则总的时间复杂度为 $O(n\log n)$。

合并过程中，使用了 n 个中间变量存储，left = (int*)malloc((nl+l)*sizeof(int))，所以空间复杂度为 $O(n)$。

推导递归式：

假设 n 个元素进行归并排序需要 $T(n)$，可以将其分割成两个分别有 $n/2$ 个元素的数组分别进行归并，也就是 $2T(n/2)$，再将这两个数组合并，需要 $O(n)$的时间复杂度，则推导公式为 $T(n) = 2T(n/2)+O(n)$。

【答案】

【问题 1】

(1) k<=r

(2) arr[k]=right[j]

(3) begin<end

(4) mergeSort(arr,mid+l,end)

【问题 2】

(5) 分治

(6) $T(n)=2T(n/2)+O(n)$

(7) $O(n\log_2 n)$

(8) $O(n)$

【问题 3】

(9) $n1+n2$

9.3 动态规划法

考核说明：本节主要介绍动态规划算法的基本思想和应用，有时会考到，要能理解。

动态规划算法通常用于求解具有某种最优性质的问题。在这类问题中，可能会有许多可行解，每个解都对应于一个值，我们希望找到具有最优值(最大值或最小值)的那个解。当然，最优解可能会有多个，动态规划算法能找出其中的一个最优解。通常可按照以下几个

步骤设计动态规划算法。

(1) 找出最优解的性质，并刻画其结构特征。

(2) 递归地定义最优解的值。

(3) 以自底向上的方式计算出最优值。

(4) 根据计算最优值时得到的信息，构造一个最优解。

对一个给定的问题，若其具有以下两个性质，则可以考虑用动态规划法来求解。

(1) 最优子结构。如果一个问题的最优解中包含其子问题的最优解，就说该问题具有最优子结构。当一个问题具有最优子结构时，意思是动态规划法可能会适用，但是此时贪心策略可能也是适用的。

(2) 重叠子问题。用来解原问题的递归算法可反复地解同样的子问题，而不是总在产生新的子问题。即当一个递归算法不断地调用同一个问题时，就说该问题包含重叠子问题。此时若用分治法递归求解，则每次遇到子问题都会视为新问题，会极大地降低算法的效率，而动态规划法总是充分利用重叠子问题，对每个子问题仅计算一次，把解保存在一个在需要时就可以查看的表中，而每次查表的时间为常数。

真题链接

【例 9-5】在求解某问题时，经过分析发现该问题具有最优子结构性质，求解过程中子问题被重复求解，则采用__(64)__算法设计策略；若定义问题的解空间，以深度优先的方式搜索解空间，则采用__(65)__算法设计策略。(2013 年 11 月真题 64～65)

(64) A. 分治　　B. 动态规划　　C. 贪心　　D. 回溯

(65) A. 动态规划　　B. 贪心　　C. 回溯　　D. 分支限界

【解析】最优子结构和高度重复性是适用动态规划方法求解的主要特征。而回溯法(探索与回溯法)是一种选优搜索法，按选优条件向前搜索，以达到目标。但当探索到某一步时，若发现原先选择并不优或达不到目标，就退回一步重新选择，这种走不通就退回再走的技术为回溯法，而满足回溯条件的某个状态的点称为“回溯点”。回溯法以深度优先方式搜索解空间，可以在搜索过程中用剪枝函数避免无效搜索。

【答案】(64)B；(65)C

【例 9-6】阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。(2014 年 11 月下午试题四)

【说明】

计算一个整数数组 a 的最长递增子序列长度的方法描述如下：

假设数组 a 的长度为 n，用数组 b 的元素 $b[i]$记录以 $a[i](0\leqslant i<n)$为结尾元素的最长递增子序列的长度为 $\max\limits_{0\leqslant i<n}\{b[i]\}$；其中 $b[i]$满足最优子结构，可递归定义为：

$$\begin{cases} b[0]=1 \\ b[i]=\max\limits_{\substack{0\leqslant k\leqslant i \\ a[k]\leqslant a[i]}}\{b[k]\}+1 \end{cases}$$

【C 代码】

下面是算法的 C 语言实现。

(1) 常量和变量说明

a：长度为 n 的整数数组，待求其最长递增子序列。

b：长度为 n 的数组，$b[i]$记录以 $a[i](0\leq i<n)$为结尾元素的最长递增子序列的长度，其中 $0\leq i<n$。

len：最长递增子序列的长度。

i,j：循环变量。

temp：临时变量。

(2) C 程序

```
#jnclude <stdio.h>
mt maxL(int*b, mt n) {
mt I, temp=0;
for(i=0; i<n; i++) {
  (b[i] >temp)
temp = b[i]
}
return temp;
}
int main() {
int n, a[100], b[100], i,j,len;
scanf("%d", &n);
for(i=0; i<n; i++) {
scanf("%d", &a[i]);
}
(1):
for(i=1;i<n; i++)  {
for(j=0,len=0;  (2)  ;j++){
if(  (3) && len <b[j])
Ien=b[j];
}
(4) ;
}
Printf("len:%d\n", maxL(b,n));
Printf("\n");
}
```

【问题 1】

根据说明和 C 代码，填充 C 代码中的空(1)～(4)。

【问题 2】

根据说明和 C 代码，算法采用了__(5)__设计策略，时间复杂度为__(6)__(用 O 符号表示)。

【问题 3】

已知数组 a= {3,10,5,15,6,8}，据说明和 C 代码，给出数组 b 的元素值。

【解析】

本题考查最长递增序列问题，是一种动态规划法，也考查时间复杂度的计算。

【答案】

【问题 1】

(1) b[0]=1

(2) j<=i

(3) a[j]<=a[i]

(4) b[i]=len+1

【问题 2】

(5) 动态规划法

(6) $O(n^2)$

【问题 3】

b={1,2,2,3,3,4}

9.4 贪　心　法

考核说明：本节主要介绍贪心算法的基本思想和应用，有时会考到，要能掌握贪心算法的 C 语言实现。

和动态规划法一样，贪心法也经常用于解决最优化问题。不过与动态规划法不同的是，贪心法在解决问题的策略上是仅根据当前已有的信息做出选择，而且一旦做出选择，不管将来有什么结果，这个选择都不会改变。换言之，贪心法并不是从整体最优考虑，它所做出的选择只是在某种意义上的局部最优。

用贪心法求解的问题一般具有两个重要的性质。

(1) 最优子结构。当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构。问题的最优子结构是该问题可以采用动态规划法或者贪心法求解的关键性质。

(2) 贪心选择性质。问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来得到。这是贪心法和动态规划法的主要区别。

典型应用：装箱问题

装箱问题可简述如下：设有编号为 $0,1,\cdots,n-1$ 的 n 种物品，体积分别为 $v_0,v_1,v_2,\cdots,v_{n-1}$，将这 n 种物品装到容量都为 V 的若干箱子中。约定这 n 种物品的体积均不超过 V。不同的装箱方案所需要的箱子数目可能不同，装箱问题要求使装完这 n 种物品的箱子数最少。

算法描述如下。

```
{
    输入箱子的容量 V;
    输入物品种类 n;
    按体积从大到小排列，输入各物品的体积;
    预置已用箱子链为空;
    预置已用箱子计数器 box_count 为 0;
    for(i = 0; i < n; i++){                //物品 i 按以下步骤装箱
        从已用的第一只箱子开始顺序寻找能放入物品 i 的箱子 j;
        if(已用箱子都不能再放物品 i){
            另用一只箱子，并将物品 i 放入该箱子;
            box_count++;
        }else{
            将物品 i 放入箱子 j 中;
        }
    }
}
```

真题链接

【例 9-7】Prim 算法和 Kruscal 算法都是无向连通网的最小生成树的算法，Prim 算法从一个顶点开始，每次从剩余的顶点中加入一个顶点，该顶点与当前的生成树中的顶点的连边权重最小，直到得到一颗最小生成树；Kruscal 算法从权重最小的边开始，每次从不在当前的生成树顶点中选择权重最小的边加入，直到得到一颗最小生成树，这两个算法都采用了<u>(64)</u>设计策略，且<u>(65)</u>。(2014 年 5 月真题 64～65)

(64) A. 分治　　B. 贪心　　C. 动态规划　　D. 回溯

(65) A. 若网较稠密，则 Prim 算法更好　　B. 两个算法得到的最小生成树是一样的

C. Prim 算法比 Kruscal 算法效率更高　　D. Kruscal 算法比 Prim 算法效率更高

【解析】Prim 算法和 Kruscal 算法都是基于贪心算法的应用。Prim 算法的时间复杂度为 $O(n^2)$，与图中边数无关，该算法适合于稠密图。Kruskal 算法的时间复杂度只和边有关系，为 $O(e\log_2 e)$，由于 Kruskal 算法只与边有关，因此适合求稀疏图的最小生成树。可见，时间复杂度为 $O(n^2)$。若问题的规模增加了 16 倍，则运行时间增加了 16^2=256 倍。

【答案】(64)B；(65)A

【例 9-8】(2013 年 5 月下午试题四)

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

设有 m 台完全相同的机器运行 n 个独立的任务，运行任务 i 所需要的时间为 t_i，要求确定一个调度方案，使得完成所有任务需要的时间最短。

假设任务已经按照其运行时间从大到小排序，算法基于最长运行时间作业优先的策略；按顺序先把每个任务分配到一台机器上，然后将剩余的任务一次放入最先空闲的机器。

【C 代码】

下面是算法的 C 语言实现。

(1) 常量和变量说明

m：机器数。

n：任务数。

t[]：输入数组，长度为 n，其中每个元素表示任务的运行时间，下标从 0 开始。

s[ ][ ]：二维数组，长度为 $m*n$，下标从 0 开始，其中元素 s[i][j]表示机器 i 运行的任务 j 的编号。

d[ ]：数组，长度为 m，其中元素 d[i]表示机器 i 的运行时间，下标从 0 开始。

count[ ]：数组，长度为 m，下标从 0 开始，其中元素 count[i]表示机器 i 运行的任务数。

i：循环变量。

j：循环变量。

k：临时变量。

max：完成所有任务的时间。

min：临时变量。

(2) 函数 schedule

```
void  schedule(){
     int i,j,k max=0;
     for(i=0;i<m;i++){
```

```
            d[i]=0;
           for(j=0;j<n;j++){
            s[i][j]=0;
        }
    }
    for(i=0;i<m;i++){                    //分配前 m 个任务
       s[i][0]=i;
    ______(1)______;
    count[i]=1;
    }
    for(_____(2)_____;i<n;i++){        //分配后 n-m 个任务
    int min=d[0];
    k=0;
    for(j=1;j<m;j++){               //确定空闲机器
    if(min>d[j]){
       min=d[j];
       k=j;                      //机器 k 空闲
    }
    }
    ______(3)______;
    count[k]=count[k]+1;
    d[k]=d[k]+t[i];

    for(i=0;i<m;i++){                //确定完成所有任务所需要的时间
        if(______(4)______){
       max=d[i];
    }
         }
         }
    }
```

【问题 1】

根据说明和 C 代码，填充 C 代码中的空(1)～(4)。

【问题 2】

根据说明和 C 代码，该问题采用了__(5)__算法设计策略，时间复杂度为__(6)__(用 O 符号表示)。

【问题 3】

考虑实例 m=3(编号 0～2)，n=7(编号 0～6)，各任务的运行时间为{16,14,6,5,4,3,2}。则在机器 0、1 和 2 上运行的任务分别为__(7)__、__(8)__和__(9)__(给出任务编号)。从任务开始运行到完成所需要的时间为__(10)__。

【解析】

本题考查算法的设计和分析技术中的贪心算法。

贪心法是一种不追求最优解，只希望得到较为满意解的方法。贪心法一般可以快速得到满意的解，因为省去了为找到最优解要穷尽所有可能而必须耗费的大量时间。贪心法常以当前情况为基础做出最优选择，而不考虑各种可能的整体情况，所以贪心法不要回溯。

【问题 1】根据上述思想和题中的说明，首先将 s[ ][ ]和 d[ ]数组初始化为 0，然后将前 m 个运行时间最长的任务分给 m 个机器，(1)中需要表示此时每个机器运行的时间，即当前已经运行的时间加上此时所运行任务的时间，可以推断(1)处为 d[i] = d[i] + t[i]，此后需将剩下的 $n-m$ 个任务按顺序分配给空闲的机器，

故(2)处将 i 初始化为以 m 为起始的任务，即 $i=m$，(3)处根据空闲的机器分配任务，所以需记录第 k 个空闲机器所运行任务的编号，即 s[k][0]=i，(4)处已经完成了任务的运行，此处需要统计所有机器所运行任务的最长时间，对于每个机器 i 的运行时间为 d[i]，存在 d[i]大于当前的最大时间 Max，就将当前机器的运行时间 d[i]赋给 Max，即 Max<d[i]。

【问题 2】根据以上分析，采用了贪心算法的策略，而时间复杂度由算法中的两个嵌套 for 循环和两个非嵌套 for 循环确定，即为 $O(2m*n+2m)$。

【问题 3】根据题中算法的思想将前三个任务分给三个机器，再将接下来的任务分给最先空闲的机器，故可知机器 0 运行任务 0，机器 1 运行任务 1、5，机器 2 运行任务 2、3、4、6；且运行的最长时间为 17。

【答案】

【问题 1】(1)d[i] = d[i] + t[i]　(2)i = m (3)s[k][0] = i　(4)Max < d[i]

【问题 2】(5)贪心　(6)$O(2m*n+2m)$

【问题 3】(7)0　(8)1、5　(9)2、3、4、6　(10)17

9.5 回 溯 法

考核说明：本节主要介绍回溯算法的基本思想和应用，是考试重点，经常会在下午考试中出现，要能掌握回溯算法的 C 语言实现。

回溯法也称为试探法，该方法首先暂时放弃关于问题规模大小的限制，并将问题的候选解按某种顺序逐一枚举和检验。当发现当前候选解不可能是解时，就选择下一个候选解；倘若当前候选解除了还不满足问题规模要求外，但满足所有其他要求时，继续扩大当前候选解的规模，并继续试探。如果当前候选解满足包括问题规模在内的所有要求时，该候选解就是问题的一个解。在回溯法中，放弃当前候选解，寻找下一个候选解的过程称为回溯；扩大当前候选解的规模，以继续试探的过程称为向前试探。

应用回溯法解问题时，首先应明确定义问题的解空间。问题的解空间应至少包含问题的一个(最优)解。

确定了解空间的组织结构后，回溯法从开始结点(根结点)出发，以深度优先的方式搜索整个解空间。这个开始结点就成为一个活结点，同时也成为当前的扩展结点。在当前的扩展结点处，搜索向纵深方向移至一个新结点。这个新结点就成为一个新的活结点，并成为当前扩展结点。如果在当前扩展结点处不能再向纵深方向移动，则当前的扩展结点就成为死结点。换句话说，这个结点不再是一个活结点。此时，应往回移动(回溯)至最近的一个活结点处，并使这个活结点成为当前的扩展结点。回溯法即以这种工作方式递归地在解空间中搜索，直至找到所要求的解或解空间中已无活结点时为止。

典型应用：*n* 后问题

n 后问题是一个源于国际象棋的问题，n 后问题要求在一个 $n×n$ 格的棋盘上放置 n 个皇后，使得它们彼此不受攻击。按照国际象棋的规则，一个皇后可以攻击与之在同一行或同一列或同一条斜线上的其他任何棋子。因此 n 后问题等价于要求在一个 $n×n$ 格的棋盘上放置 n 个皇后，使得任何 2 个皇后不能被放在同一行或同一列或同一条斜线上。

求解算法如下。

```
{
    输入棋盘大小 n;      //一个 n×n 的棋盘
    m = 0;               //从空配置开始
    good = 1;            //空配置是一个合理的情况
    do{                  //循环找解
        if(good){
            if(m == n){//找到一个解
                输出解;
                改变之，形成下一个候选解;
            }else{
                扩展当前候选解至下一列;
            }
        }else{  //回溯调整
            改变之,继续寻找候选解;
        }
    }while(m != 0);
}
```

9.6　应试加油站

9.6.1　考频统计

通过对近几年考试真题的分析，“算法与设计”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 9-1 所示。

表 9-1　历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：无		0
	下午：试题四	动态规划法及时间复杂度的计算	15
2014 年 5 月	上午：62～65	算法的时间复杂度、贪心算法	4
	下午：试题四	归并算法及时间复杂度的计算	15
2013 年 11 月	上午：64～65	动态规划算法和回溯	2
	下午：试题四	动态规划算法及其时间复杂度的计算	15
2013 年 5 月	上午：60～63	贪心算法和背包问题、分治算法	4
	下午：试题四	贪心算法及其时间复杂度的计算	15
2012 年 11 月	上午：64～65	贪心算法	2
	下午：试题四	最先适宜算法和最优适宜算法，时间复杂度计算	**15**
2012 年 5 月	上午：62～65	算法的时间复杂度和空间复杂度计算、动态规划算法	4
	下午：试题四	动态规划算法及时间复杂度计算	15

纵观历年试卷，本章知识点是以选择题和综合分析题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量为 2～4 道选择题，所占分值为 2～4(占试卷总分值 75 分中的 3%～5%)；在下午试卷中，会有 1 道综合分析题，所占分值大约为 15 分(约占试卷总分

值 75 分中的 20%)。本章试题以理论为主，难度较高。

9.6.2 解题技巧

【典型题 9-1】考虑下述背包问题的实例。有 5 件物品，背包容量为 100，每件物品的价值和重量如图 9-1 所示，并已经按照物品的单位重量价值从小到大排好序。根据物品单位重量价值大优先的策略装入背包中，则采用了 (60) 设计策略。考虑 0/1 背包问题(每件物品或者全部装入背包或者不装入背包)和部分背包问题(物品可以部分装入背包)，求解该实例得到的最大价值分别为 (61) 。(2013 年 5 月真题 60～61)

物品编号	价值	重量
1	50	5
2	200	25
3	180	30
4	225	45
5	200	50

图 9-1 物品的价值和重量

(60) A. 分治　　B. 贪心　　C. 动态规划　　D. 回溯

(61) A. 605 和 630　　B. 605 和 605　　C. 430 和 630　　D. 630 和 430

【解析】本题考查贪心算法和背包问题的知识点。

贪心算法(又称贪婪算法)是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，它所做出的仅是在某种意义上的局部最优解。贪心算法不是对所有问题都能得到整体最优解，但对范围相当广泛的许多问题它能产生整体最优解或者是整体最优解的近似解。

采用 0/1 背包考虑该问题时，只能放入 1、2、3 号物品，故总价值为 430，采用部分背包可以将物品拆分，故放入 1、2、3 号物品后还可以将编号 4 的物品部分地装入，使得背包容量尽量满，故总容量为 630。

【答案】(60)B；(61)C

【典型题 9-2】霍夫曼编码将频繁出现的字符采用短编码，出现频率较低的字符采用长编码。具体的操作过程为：i)以每个字符的出现频率作为关键字构建最小优先级队列；ii)取出关键字最小的两个节点生成子树，根节点的关键字为孩子节点关键字之和，并将根节点插入到最小优先级队列中，直至得到一棵最优编码树。霍夫曼编码方案是基于 (64) 策略的，用该方案对包含 a 到 f 六个字符的文件进行编码，文件包含 100000 个字符，每个字符的出现频率(用百分比表示)如表 9-2 所示，则与固定长度编码相比，该编码方案节省了(65) 存储空间。(2012 年 11 月真题 64～65)

表 9-2 字符出现频率

字符	a	b	c	d	e	f
出现频率	18	32	4	8	12	26

(64) A. 分治　　B. 贪心　　C. 动态规划　　D. 回溯

(65) A. 21%　　B. 27%　　C. 18%　　D. 36%

【解析】贪心算法在解决最优化问题上是仅根据当前已有的信息做出选择，即不是从整体最优考虑，它所做出的选择只是力求局部最优。本题给出的霍夫曼编码操作过程基于典型的贪心策略。

采用固定长度编码，需要 3 位二进制数字来表示 6 个字符，即 a=000，b=001，c=010，d=011，e=100，

f=101。这种方法需要 300000 位来对整个源文件编码。采用霍夫曼编码，频繁出现的字符采用短编码，出现频率较低的字符采用长编码，这种编码方式需要(32×1+26×3+18×3+12×3+4×4+8×4)×1000=248000 位。因此与固定长度编码相比，该编码方案节省的存储空间为(300000−248000)/300000 = 17.3%≈18%。

【答案】(64)B；(65)C

【典型题 9-3】某货车运输公司有一个中央仓库和 n 个运输目的地，每天要从中央仓库将货物运输到所有运输目的地，到达每个运输目的地一次且仅一次，最后回到中央仓库。在两个地点 i 和 j 之间运输货物存在费用 C_{ij}，为求解旅行费用总和最小的运输路径，设计如下算法：首先选择离中央仓库最近的运输目的地 1，然后选择离运输目的地 1 最近的运输目的地 2，……，每次在未访问过的运输目的地中选择离当前运输目的地最近的运输目的地，最后回到中央仓库。则该算法采用了__(63)__算法设计策略，其时间复杂度为__(64)__。(2012 年 5 月真题 63～64)

(63) A. 分治　　B. 动态规划　　C. 贪心　　D. 回溯

(64) A. $\Theta(n^2)$　　B. $\Theta(n)$　　C. $\Theta(n\lg n)$　　D. $\Theta(1)$

【解析】贪心算法不考虑整体情况，以当前情况为基础做出最优选择。很明显，题目中用到的是贪心算法。分值算法是将规模为 n 的问题分解为 k 个子问题，这些子问题相互独立，且与原问题相同，然后将子问题的解合并得到原问题的解。动态规划算法与分值算法类似，但分解后的子问题往往不是独立的。回溯法要在包含问题的所有解的解空间中，按照深度优先的策略，从根节点出发搜索解空间。

在选择路径时，首先选择离中央仓库最近的运输目的地 1，需要将所有 n 个目的地到中央仓库的距离进行比较，选择最近的作为目的地 1，相当于从 n 个数中选择一个最小数，此时比较了 $n-1$ 次；然后选择离目的地 1 最近的目的地 2，此时需要将其余 $n-1$ 个目的地到目的地 1 的距离进行比较，相当于从 $n-1$ 个数中选择一个最小数，此时比较了 $n-2$ 次，以此类推，共需比较 $n-1+n-2+n-3+\cdots+2+1=(n-1)(n-2)/2=(n^2-3n+2)/2$ 次，算法的时间复杂度为$\Theta(n^2)$。

【答案】(63)C；(64)A

【典型题 9-4】现要对 n 个实数(仅包含正实数和负实数)组成的数组 A 重新进行排列，使得其中所有的负实数都位于正实数之前。求解该问题的算法的伪代码如下，则该算法的时间和空间复杂度为__(65)__。(2012 年 5 月真题 65)

```
i=0;j=n-1
while i<j do
   while A[i]<0 do
      i=i+1;
   while A[j]>0 do
      j=j-1;
   if i<j do
      交换 A[i]和 A[j]
```

(65) A. $\Theta(n)$ 和 $\Theta(n)$　　B. $\Theta(1)$ 和 $\Theta(n)$　　C. $\Theta(n)$ 和 $\Theta(1)$　　D. $\Theta(1)$ 和 $\Theta(1)$

【解析】算法中用到了两个辅助变量 i 和 j，算法的空间复杂度为$\Theta(1)$。在重新排列过程中，从数组的两端进行比较，从 i=0 开始判断 A[i]是否为负数，i 为负数的时候，$i=i+1$，直到 A[i]为正数；从 $j=n-1$ 开始判断 A[i]是否为正数，如果为正数，$j=j-1$，直到 A[j]为负数。当 $i<j$ 的时候交换 A[i]和 A[j]的值，然后继续判断 A[i]和 A[j]的值。数组 A 中的元素个数为 n，A[i]<0 和 A[j]>0 的比较次数共为 $n+2$，$i=i+1$ 和 $j=j-1$ 执行的次数最多为 $n+2$ 次，if 语句中的 $i<j$ 的比较和交换 A[i]和 A[j]的操作分别最多执行 $n-1$ 次，while 循序的条件判断至多执行 n 次。可见，算法的时间复杂度为$\Theta(n)$。

【答案】C

【典型题 9-5】(2013 年 11 月下午试题 4)

阅读以下说明和 C 代码，根据要求回答问题 1～问题 3。

【说明】

某工程计算中要完成多个矩阵相乘(链乘)的计算任务。

两个矩阵相乘要求第一个矩阵的列数等于第二个矩阵的行数，计算量主要由进行乘法运算的次数决定。采用标准的矩阵相乘算法，计算 $A_{m\times n}*B_{n\times p}$，需要 $m*n*p$ 次乘法运算。

矩阵相乘满足结合律，多个矩阵相乘，不同的计算顺序会产生不同的计算量。以矩阵 $A1_{10\times100}$，$A2_{100\times5}$，$A3_{5\times50}$ 三个矩阵相乘为例，若按(A1*A2)*A3 计算，则需要进行 10*100*5+10*5*50=7500 次乘法运算；若按 A1*(A2*A3)计算，则需要进行 100*5*50+10*100*50=75000 次乘法运算。可见不同的计算顺序对计算量有很大的影响。

矩阵链乘问题可描述为：给定 n 个矩阵 $<A1,A2,\cdots,An>$，矩阵 Ai 的维数为 $p_{i-1}\times p_i$，其中 i=1,2,…,n。确定一种乘法顺序，使得这 n 个矩阵相乘时进行乘法的运算次数最少。

由于可能的计算顺序数量非常庞大，对较大的 n，用蛮力法确定计算顺序是不实际的。经过对问题进行分析，发现矩阵链乘问题具有最优子结构，即若 A1*A2*…*An 的一个最优计算顺序从第 k 个矩阵处断开，即分为 A1*A2*…*Ak 和 Ak+1*Ak+2*…*An 两个子问题，则该最优解应该包含 A1*A2*…*Ak 的一个最优计算顺序和 Ak+1*Ak+2*…*An 的一个最优计算顺序。据此构造递归式：

$$\text{cost}[i][j]=\begin{cases}0 & \text{if } i=j\\ \min_{i\le k<j}\text{cost}[i][k]+\text{cost}[k+1][j]+p_i*p_{k+1}*p_{j+1} & \text{if } i<j\end{cases}$$

其中，cost[i][j]表示 Ai+1*Ai+2*…*Aj+1 的最优计算的计算代价。最终需要求解 cost[0][n−1]。

【C 代码】

算法实现采用自底向上的计算过程。首先计算两个矩阵相乘的计算量，然后依次计算 3 个矩阵、4 个矩阵、…、n 个矩阵相乘的最小计算量及最优计算顺序。下面是该算法的 C 语言实现。

(1) 主要变量说明

n：矩阵数。

seq[]：矩阵维数序列。

cost[][]：二维数组，长度为 $n*n$，其中元素 cost[i][j]表示 Ai+1*Ai+2*…*Aj+1 的最优计算的计算代价。

trace[][]：二维数组，长度为 $n*n$，其中元素 trace[i][j]表示 Ai+1*Ai+2*…*Aj+1 的最优计算对应的划分位置，即 k。

(2) 函数 cmm

```
#define  N  100
int cost[N][N];
  int trace[N][N];
  int cmm(int n,int seq[]){
     int tempCost;
     int tempTrace;
     int i,j,k,p;
     int temp;
     for(i=0;i<n;i++){   cost[i][i]=0;   }
     for(p=1;p<n;p++){
          for(i=0;____(1)____;i++){
```

```
            (2)    ;
            tempCost=-1;
            for(k=i;k<j;k++){
                temp=   (3)   ;
                if(tempCost==-1 || tempCost>temp){
                    tempCost=temp;
                      (4)   ;
                }
            }
            cost[i][j]=tempCost;
            trace[i][j]=tempTrace;
        }
    }
    return cost[0][n-1];
}
```

【问题 1】

根据以上说明和 C 代码，填充 C 代码中的空(1)～(4)。

【问题 2】

根据以上说明和 C 代码，该问题采用了__(5)__算法设计策略，时间复杂度为__(6)__(用 O 符号表示)。

【问题 3】

考虑实例 n=6，各个矩阵的维数：$A1$ 为 5*10，$A2$ 为 10*3，$A3$ 为 3*12，$A4$ 为 12*5，$A5$ 为 5*50，$A6$ 为 50*6，即维数序列为 5,10,3,12,5,50,6。则根据上述 C 代码得到的一个最优计算顺序为__(7)__ (用加括号方式表示计算顺序)，所需要的乘法运算次数为__(8)__。

【解析】

本题考查矩阵连乘最优调度问题，是一种动态规划算法。

【问题 1】

上述算法中，第一个循环是给 n 个 cost[i][i]赋初值 0；第 2 个循环是个外循环，其循环变量 p 是矩阵连乘的规模，(p=1 时)先计算出所有规模为 2 的 cost[i,i+1]，(p=2 时)再计算出所有规模为 3 的 cost[i,i+2]，……，最后计算出来的即为我们所求的 cost[1,n−1]，所以(1)填 $i<n-p$；第 3 个循环是内循环，其循环变量 i 表示矩阵连乘的起始位置，即从 1,1+1，…，i,1+i，…，一直算到 n−1,n，所以(2)填 $j=i+p$；第四个循环用于计算 min{cost[i,j](i≤k<j)}，所以(3)填 cost[i][k]+cost[k+1][j]+seq[i]*seq[k+1]*seq[j+1]；而(4)用于追踪取得最小花费代价的 k 值，即 tempTrace=k；而每一项的计算可在 $O(1)$时间里完成。

【问题 2】

通过题目中的说明可以看出，求解问题具有最优子结构和重叠子问题，且采用自底向上的方法求解，这表明该问题采用了动态规划设计策略。

从上述 C 程序很容易分析出，程序中没有递归，存在三重循环，故该算法的时间复杂度为 $O(n^3)$。

【问题 3】

启发式的思路是先把维度最大的消掉，如 $A5$*$A6$ 相乘之后，维度 50 就没有了，所以考虑这两个矩阵先相乘；然后是 $A3$*$A4$ 相乘之后，维度 12 就没有了，所以考虑这两个矩阵相乘；接着，$A1$*$A2$ 相乘之后，维度 10 就没有了，所以考虑这两个矩阵相乘……这样可以确定相乘的顺序($A1$*$A2$)*(($A3$*$A4$)*($A5$*$A6$))，需要的计算开销分别是 5* 50* 6 = 1500，3*12 *5 = 180，5*10 *3 = 150，3*5*6=90，5*3*6=90，把上述值相加，即 1500+180+150+90+90 = 2010。

【答案】

【问题 1】(1) $i<n-p$ (2) $j=i+p$ (3) cost[i][k]+cost[k+1][j]+seq[i]*seq[k+1]*seq[j+1] (4) tempTrace=k

【问题 2】(5) 动态规划 (6) $O(n^3)$

【问题 3】(7) ($A1$*$A2$)*(($A3$*$A4$)*($A5$*$A6$)) (8) 2010

【典型题 9-6】(2012 年 5 月下午试题 4)

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

用两台处理机 A 和 B 处理 n 个作业。设 A 和 B 处理第 i 个作业的时间分别为 a_i 和 b_i。由于各个作业的特点和机器性能的关系，对某些作业，在 A 上处理时间长，而对某些作业在 B 上处理时间长。一台处理机在某个时刻只能处理一个作业，而且作业处理是不可中断的，每个作业只能被处理一次。现要找出一个最优调度方案，使得 n 个作业被这两台处理机处理完毕的时间(所有作业被处理的时间之和)最少。

算法步骤：

(1) 确定候选解上界为最短的单台处理机处理所有作业的完成时间 m：

$$m = \min(\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i)$$

(2) 用 $p(x, y, k)=1$ 表示前 k 个作业可以在 A 用时不超过 x 且在 B 用时不超过 y 时间内处理完成，则 $p(x,y,k)=p(x-a_k, y, k-1)||p(x, y-b_k, k-1)$(||表示逻辑或操作)。

(3) 得到最短处理时间为 $\min(\max(x,y))$。

【C 代码】

下面是该算法的 C 语言实现。

(1) 常量和变量说明

n：作业数。

m：候选解上界。

a：数组，长度为 n，记录 n 个作业在 A 上的处理时间，下标从 0 开始。

b：数组，长度为 n，记录 n 个作业在 B 上的处理时间，下标从 0 开始。

k：循环变量。

p：三维数组，长度为$(m+1)*(m+1)*(n+1)$。

temp：临时变量。

max：最短处理时间。

(2) C 代码

```
#include<stdio.h>
int n, m;
int a[60], b[60], p[100][100][60];
void read(){ /*输入 n、a、b，求出 m，代码略*/}
void schedule(){ /*求解过程*/
    int x,y,k;
    for(x=0;x<=m;x++){
        for(y=0;y<m;y++){
          ___(1)___
          for(k=1;k<n;k++)
              p[x][y][k]=0;
```

```
                }
            }
            for(k=1;k<n;k++){
                for(x=0;x<=m;x++){
                    for(y=0;y<=m;y++){
                        if(x - a[k-1]>=0) ___(2)___;
                        if(___(3)___)p[x][y][k]=(p[x][y][k] ||p[x][y-b[k-1]][k-1]);
                    }
                }
            }
        }
        void write(){ /*确定最优解并输出*/
            int x,y,temp,max=m;
            for(x=0;x<=m;x++){
                for(y=0;y<=m;y++){
                    if(___(4)___){
                        temp=___(5)___;
                            if(temp< max)max = temp;
                        }
                    }
            }
            printf("\n%d\n",max);
        }
        void main(){read();schedule();write();}
```

【问题 1】

根据以上说明和 C 代码，填充 C 代码中的空(1)～(5)。

【问题 2】

根据以上 C 代码，算法的时间复杂度为___(6)___(用 O 符号表示)。

【问题 3】

考虑 6 个作业的实例，各个作业在两台处理机上的处理时间如表 9-3 所示。该实例的最优解为___(7)___，最优解的值(即最短处理时间)为___(8)___。最优解用$(x_1, x_2, x_3, x_4, x_5, x_6)$表示，其中若第 i 个作业在 A 上处理，则 x_i=1，否则 x_i=2。如(1，1，1，1，2，2)表示作业 1，2，3 和 4 在 A 上处理，作业 5 和 6 在 B 上处理。

表 9-3　处理时间

	作业 1	作业 2	作业 3	作业 4	作业 5	作业 6
处理机 A	2	5	7	10	5	2
处理机 B	3	8	4	11	3	4

【解析】本题考查独立任务最优调度问题，也称双机调度，是一种动态规划算法。

【问题 1】

从 schedule()函数的第一个程序段可以看出，该段程序主要进行初始化第一个作业，下标以 0 开始，即 $p[x][y][0]$=1，内层循环里的 $p[x][y][k]$=0 用于初始化后面的 n−1 个作业。第二个程序段是对后面的 n−1 个作业，确定 $p(x, y, k)$的值。x−$a[k$−1]>=0 的判定条件若成立，则表示第 k 个作业由机器 A 处理，完成 k−1 个作业时机器 A 花费的时间是 x−$a[k$−1]，即 $p[x][y][k]$= $p[x$−$a[k$−1]][y][k−1]。(3)空要求填入一判定条件，由其后

的执行语句可知，第 k 个作业由机器 B 处理，因此判定条件应为 $y-b[k-1]>=0$。

write()程序段用于确定最优解并输出结果，即得到最短处理时间 min(max(x,y))。(4)处的判定条件是任务 n 完成，因此为 $p[x][y][n]==1$ 或其等价形式。(5)空表达式为 max(x,y)或者($x>=y$)?x:y。

【问题 2】

从程序的循环层数即可看出算法的时间复杂度。程序的最高循环层数为 3 层。最外层循环变量的变化范围是 1～n−1，次外层循环变量的变化范围是 0～m，内层循环变量的变化范围是 0～m，所以时间复杂度为 $O(m^2n)$。

【问题 3】

为了方便考生更好地理解本算法的思想，现做如下分析：

当完成 k 个作业时，设机器 A 花费了 x 时间，机器 B 所花费时间的最小值肯定是 x 的一个函数，设 $F[k][x]$表示机器 B 所花费时间的最小值，则 $F[k][x]=\text{Min}\{F[k-1][x]+b[k], F[k-1][x-a[k]]\}$。其中 $F[k-1][x]+b[k]$表示第 k 个作业由机器 B 来处理(完成 k−1 个作业时机器 A 花费的时间仍是 x)，$F[k-1][x-a[k]]$表示第 k 个作业由机器 A 处理(完成 k−1 个作业时机器 A 花费的时间是 $x-a[k]$)。

那么单个点对较大值 max(x, $F[k][x]$)，表示此时(即机器 A 花费 x 时间的情况下)所需要的总时间。而机器 A 花费的时间 x 是变化的，即 x=0,1,2,⋯,x(max)，由此构成了点对较大值序列。要求整体时间最短，取这些点对较大值序列中最小的即是。现分析前两个作业的情况：

对于第一个作业：下标以 0 开始。

首先，机器 A 所花费时间的所有可能值范围：$0\leqslant x\leqslant a[0]$。

设 $x<0$ 时，设 $F[0][x]=\infty$，则 $\max(x,\infty)=\infty$。记法意义见下。

x=0 时，$F[0][0]=3$，则 max(0,3)=3，机器 A 花费 0 时间，机器 B 花费 3 时间，而此时两个机器所需时间为 3；

x=1 时，$F[0][1]=3$，max(1,3)=3；

x=2 时，$F[0][2]=0$，则 max(2,0)=2；

那么上面的点对序列中，可以看出当 x=2 时，完成第一个作业两台机器花费最少的时间为 2，此时机器 A 花费 2 时间，机器 B 花费 0 时间。

再来看第二个作业。

首先，x 的取值范围是：$0\leqslant x\leqslant(a[0]+a[1])$。

当 $x<0$ 时，记 $F[1][x]=\infty$。之所以采用这种说法，是因为数组下标不能小于 0。$a[1]$是机器 A 完成第 2 个作业的时间，若 $x<a[1]$，则势必第 2 个作业由机器 B 来处理，即在 min()中取前者。

x=0，则 $F[1][0]=\min\{F[0][0]+b[2], F[0][0-a[1]]\}=\min\{3+8,\infty\}=11$，进而 max(0,11)=11；

x=1，则 $F[1][1]=\min\{F[0][1]+b[2], F[0][1-a[1]]\}=\min\{3+8,\infty\}=11$，进而 max(11)=11；

x=2，则 $F[1][2]=\min\{F[0][2]+b[2], F[0][2-a[1]]\}=\min\{0+8,\infty\}=8$，进而 max(2,8)=8；

x=3，则 $F[1][3]=\min\{F[0][3]+b[2], F[0][3-a[1]]\}=\min\{0+8,\infty\}=8$，进而 max(3,8)=8；

x=4，则 $F[1][4]=\min\{F[0][4]+b[2], F[0][4-a[1]]\}=\min\{0+8,\infty\}=8$，进而 max(4,8)=8；

x=5，则 $F[1][5]=\min\{F[0][5]+b[2], F[0][5-a[1]]\}=\min\{0+8,3\}=3$，进而 max(5,3)=5；

x=6，则 $F[1][6]=\min\{F[0][6]+b[2], F[0][6-a[1]]\}=\min\{0+8,3\}=3$，进而 max(6,3)=6；

x=7，则 $F[1][7]=\min\{F[0][7]+b[2], F[0][7-a[1]]\}=\min\{0+8,0\}=0$，进而 max(7,0)=7；

从上面的点对序列中，可以看出当 x=5 时，完成两个作业两台机器花费最少的时间为 5，此时机器 A

花费 5 时间，机器 B 花费 3 时间。

接下来依次类推即可，最终该实例的最优解为(1,1,2,2,1,1)，最短处理时间为 15。这里提供当各个作业完成时的最短处理时间，考生可自行推导：2，5，7，12，14，15。

【答案】

【问题 1】(1)$p[x][y][0]=1$　　(2)$p[x][y][k]= p[x-a[k-1]][y][k-1]$　　(3)$y-b[k-1]>=0$

(4)$p[x][y][n]==1$，或 $p[x][y][n]$ 或 $p[x][y][n]!=0$　　(5)$(x>=y)\ ?x:y$

【问题 2】(6)$O(m^2n)$

【问题 3】(7)(1,1,2,2,1,1)　(8)15

【典型题 9-7】(2012 年 11 月下午试题四)

阅读下列说明和 C 代码，回答问题 1 至问题 3，将解答写在答题纸的对应栏内。

【说明】

设有 n 个货物要装入若干个容重为 C 的集装箱以便运输，这 n 个货物的体积分别为{$s1, s2, \ldots, sn$}，且有 $si\leqslant C(1\leqslant i\leqslant n)$。为节省运输成本，用尽可能少的集装箱来装运这 n 个货物。

下面分别采用最先适宜策略和最优适宜策略来求解该问题。

最先适宜策略(firstfit)是首先将所有的集装箱初始化为空，对于所有货物，按照所给的次序，每次将一个货物装入第一个能容纳它的集装箱中。

最优适宜策略(bestfit)与最先适宜策略类似，不同的是，它总是把货物装到能容纳它且目前剩余容重最小的集装箱，使得该箱子装入货物后闲置空间最小。

【C 代码】

下面是这两个算法的 C 语言核心代码。

(1) 变量说明

n：货物数。

C：集装箱容量。

s：数组，长度为 n，其中每个元素表示货物的体积，下标从 0 开始。

b：数组，长度为 n，$b[i]$表示第 $n+i$ 个集装箱当前已经装入货物的体积，下标从 0 开始。

i，j：循环变量。

k：所需的集装箱数。

min：当前所用的各集装箱装入了第 i 个货物后的最小剩余容量。

m：当前所需的集装箱数。

temp：临时变量。

(2) 函数 firstfit

```
int firstfit(){
    int i,j;
    k=0;
    for(i=0;i<n;i++){
        b[i]=0;
    }
    for(i=0;i<n;i++){
          (1)  ;
        while(C-b[j]<s[i]){
```

```
            j++;
        }
          (2)  ;
        k=k>(j+1)?k:(j+1);
    }
    return k;
}
```

(3) 函数 bestfit

```
int bestfit(){
    int i,j,min,m,temp;
    k=0;
    for(i=0;i<n;i++){
        b[i]=0;
    }
    for(i=0;i<n;i++){
        min=C;
        m=k+1;
        for(j=0;j<k+1;j++){
            temp=C-b[j]-s[i];
            if(temp>0 && temp<min){
                  (3)  ;
                m=j;
            }
        }
          (4)  ;
        k=k>(j+1)?k:(j+1);
    }
    return k;
}
```

【问题 1】

根据【说明】和【C 代码】，填充 C 代码中的空(1)～(4)。

【问题 2】

根据【说明】和【C 代码】，该问题在最先适宜和最优适宜策略下分别采用了 (5) 和 (6) 算法设计策略，时间复杂度分别为 (7) 和 (8) (用 O 符号表示)。

【问题 3】

考虑实例 n=10，C=10，各个货物的体积为{4,2,7,3,5,4,2,3,6,2}。该实例在最先适宜和最优适宜策略下所需的集装箱分别为 (9) 和 (10) 。考虑一般的情况，这两种求解策略能否确保得到最优解？ (11) (能或否)

【解析】

本题考查算法的 C 语言实现、时间复杂度的计算及应用，试题本身具有一定的难度。

【问题 1】

本题描述的算法包括最先适宜算法和最优适宜算法。其中，最先适宜算法要求按顺序给货物找到一个能容纳它的集装箱，找到即可装箱。这里的关键在于找到第一个能容纳它的集装箱，从头到尾遍历各集装箱即可。firstfit 函数用于实现最先适宜算法。定位到空(1)处，其上面的 for 循环用于对所有 n 个货物进行

遍历，分别找出满足条件 $C-b[j]>=s[i]$ 的集装箱。但条件 $C-b[j]<s[i]$ 中的变量 j 在空(1)前并没有显式的赋值语句，且遍历各集装箱应从第一个开始，因此空(1)应填入 j=0。空(2)处表示货物已放入集装箱的情况，应更新装入货物后的体积，因此空(2)应填入 $b[j]= b[j]+s[i]$。

最优适宜算法不仅要找到能容纳货物的集装箱，而且还要求该集装箱的剩余容量最小。bestfit 函数用于实现最优适宜算法。该函数的 for 循环语句中的 temp 表示剩余的最小容量，若其小于 min，则应更新其值。因此，空(3)应填入 min=temp。和 firstfit 函数中空(2)类似的思路，空(4)应填入 $b[m]= b[m]+s[i]$。

【问题 2】

贪心算法在解决最优化问题上是仅根据当前已有的信息作出选择，即不是从整体最优考虑，它所做出的选择只是力求局部最优。最先适宜策略和最优适宜策略均采用了该算法设计策略。

对于时间复杂度，应根据程序中循环的层数及每层循环的次数来进行计算。可以很容易地判断，这两种算法的时间复杂度均为 $O(n^2)$。

【问题 3】

本问题考查对程序的具体理解和应用。

对 firstfit 函数进行遍历的结果如下：

i	$b[j]$	k
0	$b[0]$=4	1
1	$b[0]$=6	1
2	$b[1]$=7	2
3	$b[1]$=10	2
4	$b[2]$=5	3
5	$b[2]$=9	3
6	$b[3]$=2	4
7	$b[3]$=5	4
8	$b[4]$=6	5
9	$b[4]$=8	5

因此，该实例在最先适宜策略下所需的集装箱数为 5。同理可对 bestfit 函数进行遍历，可得到该实例在最优适宜策略下所需的集装箱数为 4，遍历过程可由考生自己设计，以增强对整个算法的理解。

由于贪心算法在解决最优化问题上是仅根据当前已有的信息做出选择，即不是从整体最优考虑，它所做出的选择只是力求局部最优，因此这两种求解策略不能确保得到最优解。

【答案】

【问题 1】

(1)j=0　　(2)$b[j]= b[j]+s[i]$　　(3)min=temp　　(4)$b[m]= b[m]+s[i]$

【问题 2】

(5)贪心　　(6)贪心　　(7)$O(n^2)$　　(8)$O(n^2)$

【问题 3】

(9) 5 (10) 4 (11) 否

9.7 过关习题

1. 对 n 个元素值分别为−1、0 或 1 的整型数组 A 进行升序排序的算法描述如下：统计 A 中−1、0 和 1 的个数，设分别为 n_1、n_2 和 n_3，然后将 A 中的前 n_1 个元素赋值为−1，第 n_1+1 到 n_1+n_2 个元素赋值为 0，最后 n_3 个元素赋值为 1。该算法的时间复杂度和空间复杂度分别为______。

   A. $\Theta(n)$和$\Theta(1)$　　B. $\Theta(n)$和$\Theta(n)$

   C. $\Theta(n^2)$和$\Theta(1)$　　D. $\Theta(n^2)$和$\Theta(n)$

2. 下面 C 程序段中 count++语句执行的次数为______。

```
for(int i=1;i<=11;i*=2)
      for(int j=1;j<=i;j++)
           count++;
```

   A. 15　　B. 16　　C. 31　　D. 32

3. 设算法 A 的时间复杂度可用递归式 $T(n)=\begin{cases}\Theta(1), & n=1\\ 7T(n/2)+n^2, & n>1\end{cases}$ 表示，算法 B 的时间复杂度可用递归式 $T(n)=\begin{cases}\Theta(1), & n=1\\ aT(n/4)+n^2, & n>1\end{cases}$ 表示，若要使得算法 B 渐进地快于算法 A，则 a 的最大整数为______。

   A. 48　　B. 49　　C. 13　　D. 14

4. 某算法的时间复杂度可用递归式 $T(n)=\begin{cases}\Theta(1) & n=\\ 6T(n/5)+n, n>1\end{cases}$ 表示，若用 Θ 表示，则正确的是______。

   A. $\Theta(n^{\log_5 6})$　　B. $\Theta(n^2)$　　C. $\Theta(n)$　　D. $\Theta(n^{\log_6 5})$

5. 某算法的时间复杂度可用递归式 $T(n)=\begin{cases}\Theta(1) & ,n=1\\ 2T(n/2)+n\lg n, n>1\end{cases}$ 表示，若用 Θ 表示该算法的渐近时间复杂度的紧致界，则正确的是______。

   A. $\Theta(n\lg^2 n)$　　B. $\Theta(n\lg n)$

   C. $\Theta(n^2)$　　D. $\Theta(n^3)$

6. 在有 n 个无序无重复元素值的数组中查找第 i 小的数的算法描述如下：任意取一个元素 r，用划分操作确定其在数组中的位置，假设元素 r 为第 k 小的数。若 i 等于 k，则返回该元素值；若 i 小于 k，则在划分的前半部分递归进行划分操作找第 i 小的数；否则在划分的后半部分递归进行划分操作找第 $k-i$ 小的数。该算法是一种基于______策略的算法。

   A. 分治　　B. 动态规划　　C. 贪心　　D. 回溯

7. 分治算法设计技术______。

   A. 一般由三个步骤组成：问题划分、递归求解、合并解

   B. 一定是用递归技术来实现

   C. 将问题划分为 k 个规模相等的子问题

   D. 划分代价很小而合并代价很大

8. 用动态规划策略求解矩阵连乘问题 $M_1 \times M_2 \times M_3 \times M_4$，其中 $M_1(20\times5)$、$M_2(5\times35)$、$M_3(35\times4)$和 $M_4(4\times25)$，则最优的计算次序为______。

A. $((M_1 \times M_2) \times M_3) \times M_4$　　B. $(M_1 \times M_2) \times (M_3 \times M_4)$

C. $(M_1 \times (M_2 \times M_3)) \times M_4$　　D. $M_1 \times (M_2 \times (M_3 \times M_4))$

9. 迪杰斯特拉(Dijkstra)算法用于求解图上的单源点最短路径。该算法按路径长度递增次序产生最短路径，本质上说，该算法是一种基于______策略的算法。

A. 分治　　B. 动态规划　　C. 贪心　　D. 回溯

10. ______不能保证求得 0-1 背包问题的最优解。

A. 分支限界法　　B. 贪心算法　　C. 回溯法　　D. 动态规划策略

11. 要在 8×8 的棋盘上摆放 8 个“皇后”，要求“皇后”之间不能发生冲突，即任何两个“皇后”不能在同一行、同一列和相同的对角线上，则一般采用______来实现。

A. 分治法　　B. 动态规划法　　C. 贪心法　　D. 回溯法

12. 阅读下列说明和 C 代码，将应填入(*n*)处的字句写在答题纸的对应栏内。

【说明】

设某机器由 *n* 个部件组成，每一个部件都可以从 *m* 个不同的供应商处购得。供应商 *j* 供应的部件 *i* 具有重量 W_{ij} 和价格 C_{ij}。设计一个算法，求解总价格不超过上限 cc 的最小重量的机器组成。

采用回溯法来求解该问题。

首先定义解空间。解空间由长度为 *n* 的向量组成，其中每个分量取值来自集合{1，2，…，*m*}，将解空间用树形结构表示。

接着从根结点开始，以深度优先的方式搜索整个解空间。从根结点开始，根结点成为活结点，同时也成为当前的扩展结点。向纵深方向考虑第一个部件从第一个供应商处购买，得到一个新结点。判断当前的机器价格(C_{11})是否超过上限(cc)，重量(W_{11})是否比当前已知的解(最小重量)大，若是，应回溯至最近的一个活结点；若否，则该新结点成为活结点，同时也成为当前的扩展结点，根结点不再是扩展结点。继续向纵深方向考虑第二个部件从第一个供应商处购买，得到一个新结点。同样判断当前的机器价格($C_{11}+C_{21}$)是否超过上限(cc)，重量($W_{11}+W_{21}$)是否比当前已知的解(最小重量)大。若是，应回溯至最近的一个活结点；若否，则该新结点成为活结点，同时也成为当前的扩展结点，原来的结点不再是扩展结点。以这种方式递归地在解空间中搜索，直到找到所要求的解或者解空间中已无活结点为止。

【C 代码】

下面是该算法的 C 语言实现。

(1) 变量说明。

n：机器的部件数。

m：供应商数。

cc：价格上限。

w[][]：二维数组，*w*[*i*][*j*]表示第 *j* 个供应商供应的第 *i* 个部件的重量。

c[][]：二维数组，*c*[*i*][*j*]表示第 *j* 个供应商供应的第 *i* 个部件的价格。

bestW：满足价格上限约束条件的最小机器重量。

bestC：最小重量机器的价格。

bestX[]：最优解，一维数组，bestX[*i*]表示第 *i* 个部件来自哪个供应商。

cw：搜索过程中机器的重量。

cp：搜索过程中机器的价格。

x[]：搜索过程中产生的解，*x*[*i*]表示第 *i* 个部件来自哪个供应商。

i：当前考虑的部件，从 0 到 *n*−1。

j：循环变量。

(2) 函数 backtrack。

```
    int n=3;
    int m=3;

    int cc=4;
    int w[3][3]={{1,2,3},{3,2,1},{2,2,2}};
    int c[3][3]={{1,2,3},{3,2,1},{2,2,2}};
    int bestW=8;
    int bestC=0;
    int bestX[3]={0,0,0};
    int cw=0;
    int cp=0;
    int x[3]={0,0,0};
  int backtrack(int i){
      int j=0;
      int found=0;
      if(i>n-1){/*得到问题解*/
          bestW= cw;
          bestC= cp;
          for(j=0;j<n;j++){
            ___(1)___;
        }
          return 1;
     }
     if(cp<=cc){/*有解*/
          found=1;
     }
     for(j=0; ___(2)___;j++){
        /*第 i 个部件从第 j 个供应商购买*/
         ___(3)___;
         cw=cw+w[i][j];
         cp=cp+c [i][j];
         if(cp<=cc &&  ___(4)___{/*深度搜索，扩展当前结点*/
              if(backtrack(i+1)){found=1;}
         }
         /*回溯*/
         cw= cw -w[i][j];
           ___(5)___;
     }
     return found;
  }
```

第 10 章

面向对象技术

本章要点

- 面向对象开发概念(类、对象、属性、封装性、继承性、多态性、对象之间的引用)
- 面向对象的分析与设计
- UML 描述方法
- 设计模式
- 面向对象程序设计语言(如 C++、Java、Visual C++)的基本机制

10.1　面向对象的基本概念

考核说明：本节主要介绍面向对象的基本概念，经常会考到，要掌握对象、类、继承、多态、封装等概念。

Peter Coad 和 Edward Yourdon 提出用下面的等式识别面向对象方法：

面向对象=对象(Object)+分类(Classification)+继承(Inherritance)+通过消息的通信(Communication with Messages)

(1) 对象。在面向对象的系统中，对象是基本的运行时的实体，它既包括数据(属性)，也包括作用于数据的操作(行为)，所以一个对象把属性和行为封装为一个整体。封装是一种信息隐蔽技术，它的目的是使对象的使用者和生产者分离，使对象的定义和实现分开。从程序设计者来看，对象是一个程序模块；从用户来看，对象为他们提供了所希望的行为。在对象内的操作通常叫作方法。一个对象通常可由对象名、属性和操作三部分组成。

(2) 消息。对象之间进行通信的一种构造称为消息。但一个消息发送给某个对象时，包含要求接收对象去执行某些活动的信息，接收到消息的对象经过解释，然后予以响应，这种通信机制称为消息传递。发送消息的对象不需要知道接收消息的对象如何对请求予以响应。

(3) 类。一个类定义了一组大体上相似的对象，所包含的方法和数据描述一组对象的共同行为和属性。类是在对象之上的抽象，对象是类的具体化，是类的实例。通常把一个类和这个类的所有对象称为“类及对象”或对象类。

(4) 继承。继承是父类和子类之间共享数据和方法的机制。这是类之间的一种关系，在定义和实现一个类的时候，可以在一个已经存在的类的基础上来进行修改，把这个已经存在的类所定义的内容作为自己的内容，并加入若干新的内容。

一个父类可以有多个子类，这些子类都是父类的特例，父类描述了这些子类的公共属性和操作。一个子类可以继承它的父类(或祖先类)中的属性和操作，这些属性和操作在子类中不必定义，子类中还可以定义自己的属性和操作。

(5) 多态。不同的对象收到同一消息可以产生完全不同的结果，这一现象叫作多态。在使用多态的时候，用户可以发送一个通用的消息，而实现的细节则由接收对象自行决定，这样，把具有通用功能的消息存放在高层次，而把不同的实现这一功能的行为放在较低层次，在这些低层次上生成的对象能够给通用消息以不同的响应。

(6) 动态绑定。绑定是一个把过程调用和响应调用所需要执行的代码加以结合的过程。在一般的程序设计语言中，绑定是在编译时进行的，称为静态绑定。动态绑定则是在运行时进行的，因此，一个给定的过程调用和代码的结合是到调用发生时才进行的。

动态绑定是和类的继承以及多态相联系的。在继承关系中，子类是父类的一个特例，所以父类对象可以出现的地方，子类对象也可以出现。因此在运行过程中，当一个对象发送消息请求服务时，要根据接收对象的具体情况将请求的操作与实现的方法进行连接，即动态绑定。

真题链接

【例 10-1】类__(37)__之间存在着一般和特殊的关系。(2014 年 11 月真题 37)

(37) A. 汽车与轮船　　B. 交通工具与飞机　　C. 轮船与飞机　　D. 汽车与飞机

【解析】一般是统称，特殊是特定具体的意思。

【答案】B

【例 10-2】多态分为参数多态、包含多态、过载多态和强制多态四种不同形式，其中__(38)__多态在许多语言中都存在，最常见的例子就是子类型化。(2014 年 11 月真题 38)

(38) A. 参数　　B. 包含　　C. 过载　　D. 强制

【解析】多态分为两种：通用的多态和特定的多态。两者的区别是前者对工作的类型不加限制，允许对不同类型的值执行相同的代码；后者只对有限数量的类型有效，而对不同类型的值可能要执行不同的代码。

通用的多态又分为参数多态(parametric)和包含多态(Inclusion)；特定的多态分为过载多态(overloading)和强制多态(coercion)。

强制多态：编译程序通过语义操作，把操作对象的类型强行加以变换，以符合函数或操作符的要求。程序设计语言中基本类型的大多数操作符，在发生不同类型的数据进行混合运算时，编译程序一般都进行强制多态。

过载(overloading)多态：同一个名(操作符、函数名)在不同的上下文中有不同的类型，程序设计语言中基本类型的大多数操作符都是过载多态的。

参数多态：采用参数化模板，通过给出不同的类型参数，使得一个结构有多种类型。

包含多态：同样的操作可用于一个类型及其子类型(注意是子类型，不是子类)。包含多态一般需要进行运行时的类型检查。

【答案】B

【例 10-3】在面向对象程序设计语言中，对象之间通过__(39)__方式进行通信。以下关于好的面向对象程序设计语言的叙述中，不正确的是__(40)__。(2014 年 11 月真题 39～40)

(39) A. 消息传递　　B. 继承　　C. 引用　　D. 多态

(40) A. 应该支持被封装的对象　　B. 应该支持类写实例的概念

C. 应该支持通过指针进行引用　　D. 应该支持继承和多态

【解析】对象间通过接口传递消息，实现通信。“继承”、“引用”、“多态”均为实现概念。“应该支持通过指针进行引用”只针对部分语言，如 C++，不具有代表性。

【答案】(39)A；(40)C

【例 10-4】某些程序设计语言中，在运行过程中当一个对象发送消息请求服务时，根据接收对象的具体情况将请求的操作与实现的方法进行连接，称为__(38)__。(2014 年 5 月真题 38)

(38) A. 静态绑定　　B. 通用绑定　　C. 动态绑定　　D. 过载绑定

【解析】所谓静态绑定是指在程序编译过程中，把函数(方法或者过程)调用与响应调用所需的代码结合的过程；动态绑定是指在执行期间判断所引用对象的实际类型，根据其实际的类型调用其相应的方法。

【答案】C

【例 10-5】在面向对象技术中，不同的对象在收到同一消息时可以产生完全不同的结果，这一现象称

为__(39)__，它由__(40)__机制来支持。利用类的层次关系，把具有通用功能的消息存放在高层次，而不同的实现这一功能的行为放在较低层次，在这些低层次上生成的对象能够给通用消息以不同的响应。(2014 年 5 月真题 39～40)

(39) A. 绑定　　B. 继承　　C. 消息　　D. 多态

(40) A. 绑定　　B. 继承　　C. 消息　　D. 多态

【解析】多态性是同一操作作用于不同的类的实例，将产生不同的执行结果，即不同类的对象收到相同的消息时，得到不同的结果。在运行时，可以通过指向基类的指针，来调用实现派生类中的方法。多态是面向对象程序设计的重要特征之一，是扩展性在“继承”之后的又一重大表现。如果一个语言只支持类而不支持多态，只能说明它是基于对象的，而不是面向对象的。

【答案】(39)D；(40)B

10.2　面向对象程序设计

考核说明：本节主要介绍面向对象设计语言和 OOP 机制，虽然考试中很少会考到，但仍需要理解 OOP 机制。

面向对象程序设计(Object Oriented Programming，OOP)的实质是选用一种面向对象程序设计语言(OOPL)，采用对象、类及其相关概念所进行的程序设计。

10.2.1　面向对象程序设计语言

1. Smalltalk

Smalltalk 并不是一种单纯的程序设计语言，而是反映面向对象程序设计思想的程序设计环境。这个系统强调了对象概念的归一性，引入了类、方法、实例等概念和术语，应用了单重继承和动态绑定，成为面向对象程序设计语言(OOPL)发展过程中的一个引人注目的里程碑。

2. Eiffel

Eiffel 的主要特点是全面的静态类型化、有大量的开发工具、支持多继承。Eiffel 也全面支持面向对象的概念。

3. C++

C++语言是一种面向对象的强类型语言，由 AT&T 的 Bell 实验室于 1980 年推出。C++语言是 C 语言的一个向上兼容的扩充，而不是一种新语言。C++是一种支持多范型的程序设计语言，它既支持面向对象的程序设计，也支持面向过程的程序设计。C++支持基本的面向对象概念：对象、类、方法、消息、子类和继承。

4. Java

Java 语言起源于 Oak 语言，Oak 语言能运行在设备的嵌入芯片上。Java 编译成伪代码，这需要一个虚拟机来对其进行解释，Java 的虚拟机几乎在每一种平台上都可以运行。这实

质上可以使得开发与机器独立无关，并且提供了通用的可移植性。

Java 把类的概念和接口的概念区分开，并试图通过只允许接口的多继承来克服多继承的危险。Java Beans 是组件，即类和其所需资源的集合，它们主要被设计用来提供定制的 GUI 小配件。

Java 中关于面向对象概念的术语有对象、类、方法、实例、变量、消息、子类和继承。

真题链接

【例 10-6】属于面向对象、解释型程序设计语言的是(18)。(2014 年 11 月真题 18)

(18) A. XML　　B. Python　　C. Prolog　　D. C++

【解析】XML：是一种用于标记电子文件使其具有结构性的标记语言。

C++：在 C 语言的基础上发展起来的，主要增加了类的功能，使其成为面向对象的程序设计语言。

Prolog：以特殊的逻辑推理形式回答用户的查询，经常用于数据库和专家系统。

Python：一种面向对象，解释型计算机程序设计语言。Python 的语法简洁而清晰，具有丰富和强大的类库。它能够把用其他语言制作的各种模块(尤其是 C/C++)很轻松地联结在一起。

【答案】B

10.2.2　面向对象程序设计语言中的 OOP 机制

1. 类

类具有实例化功能，包括实例生成和实例消除。类的实例化决定了类及其实例具有下面的特征。

- 同一个类的不同实例具有相同的数据结构，承受的是同一方法集合所定义的操作，因而具有规律相同的行为。
- 同一个类的不同实例可以持有不同的值，因而可以具有不同的状态。
- 实例的初始状态(初值)可以在实例化时确定。

2. 继承和类层次结构

当执行一个子类的实例生成方法时，首先在类层次结构中从该子类沿继承路径上溯至它的一个基类，然后自顶向下地执行该子类的所有父类的实例生产方法，最后执行该子类实例生产方法的函数体。当执行一个子类的实例消除方法时，顺序正好与之相反：先执行该子类的实例消除方法，再沿继承路径自底向上地执行该子类所有父类的实例消除方法。

类的实例化过程是一种实例的合成过程，而不仅仅是根据单个类型进行的空间分配、初始化和联编。指导编译程序进行这种合成的，就是类层次结构。

3. 对象、消息传递和方法

对象是类的实例。尽管对象的表示在形式上与一般数据结构十分相似，但是它们之间存在本质区别：对象之间通过消息传递方式进行通信。

消息传递原是一种与通信有关的概念，OOP 使得对象具有交互能力的主要模型就是消息传递模型。对象被看成用传递消息的方式互相联系的通信实体，它们既可以接收，也可

以拒绝外界发来的消息。一般情况下，对象接收它能够识别的消息，拒绝不能识别的消息。

4. 对象自身引用

对象自身引用是面向对象程序设计语言(OOPL)中的一种特有结构。这种结构在不同的 OOPL 中有不同的名称，在 C++和 Java 中称为 this，在 Smalltalk-80、Object-C 和其他一些 OOPL 中则称为 self。

对象自身引用的值和类型分别扮演了两种意义的角色：对象自身引用的值使得方法体中引用的成员名与特定的对象相关，对象自身引用的类型则决定了方法体被实际共享的范围。

5. 重置

重置的基本思想是：通过一种动态绑定机制的支持，使得子类在继承父类界面定义的前提下，用适合于自己要求的实现去置换父类中的相应实现。

在 OOPL 中，重置机制有相应的语法供开发人员选择使用。C++通过虚拟函数的定义来进行重置的声明，通过虚拟函数跳转表结构来实现重置方法体的动态绑定。Java 通过抽象方法进行重置的声明，通过方法查找实现重置方法体的动态绑定。

6. 类属类

类属是程序设计语言中普遍注重的一种参数多态机制。类属类可以看作类的模板。一个类属类是关于一组类的一个特性抽象，它强调的是这些类的成员特征中与具体类型无关的那些部分，而与具体类型相关的部分则用变元来表示。这就使得对类的集合也可以按照特性的相似性再次进行划分。类属类的一个重要作用，就是对类库的建设提供强有力的支持。

7. 无实例的类

要创建无实例的类需要语言的支持。在 C++和 Java 语言中，抽象类就是这样的类。在 C++中通过在类中定义纯虚拟函数来创建一个抽象类；在 Java 中通过在类中定义抽象方法来创建一个抽象类，或者直接将一个类声明为抽象类。

10.3 面向对象开发技术

考核说明：本节主要介绍面向对象开发技术，包括面向对象分析、设计和测试等内容，很少考到，了解即可。

(1) 面向对象分析。面向对象分析的目的是获得对应用问题的理解。理解的目的是确定系统的功能、性能要求。面向对象分析包含五个活动：认定对象、组织对象、描述对象间的相互作用、定义对象的操作和定义对象的内部信息。

(2) 面向对象设计。面向对象设计的含义是设计分析模型和实现相应源代码，在目标代码环境中这种源代码可被执行。设计期间必须充分考虑系统的稳定性，这会影响系统的结构。

对象标识期间的目标是分析对象，设计过程也是发现对象的过程，称为再处理。对象可以用预先开发的源代码实现，称这样的部分为构件。

(3) 面向对象测试。一般来说，对面向对象软件的测试可分为下列四个层次进行。

① 算法层。测试类中定义的每个方法，基本上相当于传统软件测试中的单元测试。

② 类层。测试封装在同一个类中的所有方法与属性之间的相互作用。在面向对象软件中类是基本模块，因此可以认为这是面向对象测试中所特有的模块测试。

③ 模板层。测试一组协同工作的类之间的相互作用。大体上相当于传统软件测试中的集成测试，但是也有面向对象软件的特点(例如，对象之间通过发送消息相互作用)。

④ 系统层。把各个子系统组装成完整的面向对象的软件系统，在组装过程中同时进行测试。

10.4　面向对象分析与设计

考核说明：本节主要介绍面向对象分析与设计方法：OOA 和 OOD 法、Booch 的 OOD 法和 OMT 法，需要了解各种方法的步骤。

10.4.1　OOA 和 OOD 法

1. OOA

OOA(面向对象分析)模型由下列五个层次和五个活动组成。

(1) 五个层次：主题层、对象类层、结构层、属性层、服务层。

(2) 五个活动：标志对象类、标志结构、定义主题、定义属性、定义服务。

在这种方法中定义了两种对象类之间的结构，一种称为分类结构，另一种称为组装结构。

2. OOD

OOA 中的五个层次和五个活动继续贯穿在 OOD(面向对象设计)过程中。OOD 模型由四个部分和四个活动组成。四个活动是设计问题域部件、设计人机交互部件、设计任务管理部件、设计数据管理部件。

10.4.2　Booch 的 OOD 法

Booch 认为软件开发是一个螺旋上升的过程，在螺旋上升的每个周期中有以下步骤。

(1) 标识类和对象。

(2) 确定它们的含义。

(3) 标志它们之间的关系。

(4) 说明每一个类的界面和实现。

除了类图、对象图、模块图、进程图外，Booch 的 OOD 中还使用了两种动态描述图：一种是刻画特定类实例的状态转换图，另一种是描述对象间事件变化的时序图。

10.4.3　OMT 法

对象建模技术(OMT)定义了三种模型：对象模型、动态模型和功能模型。OMT 方法有四个步骤：分析、系统设计、对象设计和实现。

1. 对象模型、动态模型和功能模型

1)　对象模型

OMT 的对象模型中除了对象、类、继承外，还有一些其他的概念。

- 链和关联：链表示实例对象间的物理或概念上的连接。关联描述具有公共结构和公共语义的一组链。
- 泛化：泛化是一个类与它的一个或多个细化类之间的关系，即一般与特殊的关系。被细化的类称为父类，每个细化的类称为子类，子类可以继承父类的特性。
- 聚集：聚集是一种整体与部分的关系，在这种关系中表示整体的对象与表示部分的对象关联。
- 模块：模块是组和类、关联和泛化的一种逻辑结构，模块给出了某个主题的视图。

2)　动态模型

动态模型描述与时间和操作顺序有关的系统特征——激发事件、事件序列、确定事件先后关系以及事件和状态的组织。

3)　功能模型

功能模型描述与值的变换有关的系统特征——功能、映射、约束和函数依赖。

2. OMT 的步骤

(1)　分析：目的是建立可理解的现实世界模型。

(2)　系统设计：确定整个系统的体系结构，形成求解问题和建立解答的高层次策略。

(3)　对象设计：建立基于分析模型的设计模型，并考虑实现的细节，设计人员根据系统设计期间建立的策略把实现细节加入到设计模型中。

(4)　实现：将对象设计阶段开发的对象类及其关系转换成特定的程序设计语言、数据库或硬件的实现。

10.5　UML 概述

考核说明：本节主要介绍 UML 的三大构造块——事物、关系和图。UML 的四种联系、UML 的各种视图是上下午试卷中的必考内容，要重点理解掌握。

10.5.1　事物

统一建模语言(UML)是面向对象软件的标准化建模语言。UML 由三个要素构成：UML 的基本构造块、支配这些构造块如何放置在一起的规则和运用于整个语言的一些公共机制。

UML 的词汇表包含三种构造块：事物、关系和图。事物是对模型中最具代表性的成分的抽象，关系是把事物结合在一起，图则聚集了相关的事物。

事物包括结构事物、行为事物、分组事物和注释事物。

(1) 结构事物。结构事物是 UML 模型中的名词。它们通常是模型的静态部分，描述概念或物理元素。结构事物包括类、接口、协作、用例、主动类、构件和结点。

(2) 行为事物。行为事物是 UML 模型的动态部分。它们是模型中的动词，描述了跨越时间和空间的行为。共有两类主要的行为事物：交互和状态机。

(3) 分组事物。分组事物是 UML 模型的组织部分。它们是一些由模型分解成的“盒子”。在所有的分组事物中，最主要的分组事物是包。

(4) 注释事物。注释事物是 UML 模型的解释部分。这些注释事物用来描述、说明和标注模型的任意元素。注解是一种主要的注释事物。注解是一个依附于一个元素或者一组元素之上，对它进行约束或解释的简单符号。

真题链接

【例 10-7】UML 中有 4 种事物：结构事物、行为事物、分组事物和注释事物。类、接口、构件属于 (41) 事物；依附于一个元素或一组元素之上对其进行约束或解释的简单符号为 (42) 事物。(2014 年 11 月真题 41～42)

(41) A. 结构　　B. 行为　　C. 分组　　D. 注释

(42) A. 结构　　B. 行为　　C. 分组　　D. 注释

【解析】模型中的基本成员。UML中包括结构事物、行为事物、分组事物和注释事物。

(1) 结构事物：模型中的静态部分。包括：类(Class)、接口(Interface)、协作(Collaboration)、用例(UseCase)、活动类、组件(Commponent)、结点(Node)。

(2) 行为事物：模型中的动态部分。包括：交互、状态机、活动。

(3) 分组事物：可以把分组事物看成是一个“盒子”，模型可以在其中被分解。目前只有一种分组事物，即包(Package)、结构事物、动作事物、甚至分组事物都有可能放在一个包中。包纯粹是概念上的，只存在于开发阶段，而组件在运行时存在。

(4) 注释事物：注释事物是 UML 模型的解释部分。

【答案】(41)A；(42)D

10.5.2　关系

UML 中有四种关系：依赖、关联、泛化和实现。各种关系图例如图 10-1 所示。

(1) 依赖(Dependency)。依赖是两个事物间的语义关系，其中一个事物(独立事物)发生变化会影响另一个事物(依赖事物)的语义。

(2) 关联(Association)。关联是一种结构关系，它描述了一组链，链是对象之间的连接。聚集(Aggregation)是一种特殊类型的关联，它描述了整体和部分间的结构关系。

(3) 泛化(Generalization)。泛化是一种特殊与一般的关系，特殊元素(子元素)的对象可替代一般元素(父元素)的对象。用这种方法，子元素共享了父元素的结构和行为。

(4) 实现(Realization)。实现是类元之间的语义关系，其中一个类元指定了由另一个类

元保证执行的契约。在两种地方要遇到实现关系：一种是在接口和实现它们的类或构件之间；另一种是在用例和实现它们的协作之间。

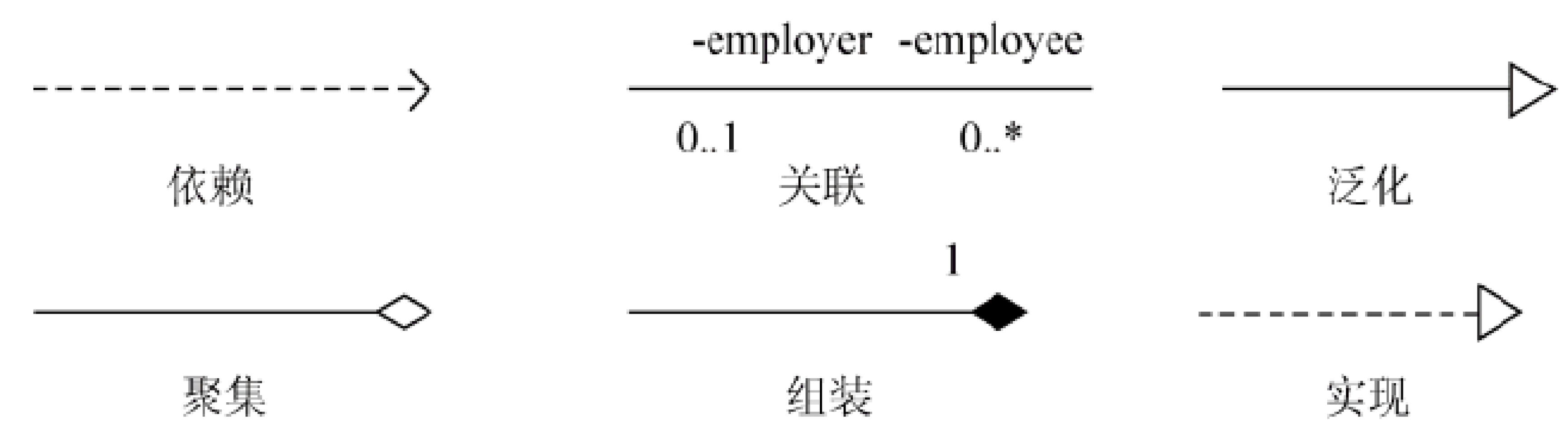


图 10-1　各种关系图例

真题链接

【例 10-8】<u>(37)</u>是一个类与它的一个或多个细化类之间的关系，即一般与特殊的关系。(2014 年 5 月真题 37)

(37) A. 泛化　　B. 关联　　C. 聚集　　D. 组合

【解析】泛化表示类与类之间的继承关系，接口与接口之间的继承关系，或类对接口的实现关系。一般泛化关系是从子类指向父类的。

对于两个相对独立的对象，当一个对象的实例与另一个对象的一些特定实例存在固定的对应关系时，这两个对象之间为关联关系。关联体现的是两个类或者类与接口之间语义级别的一种强依赖关系，这种关系一般是长期性的，而且双方的关系一般是平等的。关联可以是单向、双向的。

聚合是关联关系的一种特例，体现的是整体与部分的关系，即 has-a 的关系，此时整体与部分之间是可分离的，它们可以具有各自的生命周期，部分可以属于多个整体对象，也可以为多个整体对象共享。

组合也是关联关系的一种特例，体现的是一种 contains-a 的关系，这种关系比聚合更强，也称为强聚合；它同样体现整体与部分间的关系，但此时整体与部分是不可分的，整体的生命周期结束也就意味着部分的生命周期结束。

【答案】A

10.5.3　UML 中的图

UML 中的各种视图及其主要概念如表 10-1 所示。

表 10-1　UML 中的各种视图及其主要概念

主要的域	视　图	图	主要概念
结构	静态视图	类图	类、关联、泛化、依赖关系、实现、接口
	用例视图	用例图	用例、参与者、关联、扩展、包括、用例泛化
	实现视图	构件图	构件、接口、依赖关系、实现
	部署视图	部署图	结点、构件、依赖关系、实现
动态	状态机视图	状态机图	状态、事件、转换、动作
	活动视图	活动图	状态、活动、完成转换、分叉、结合
	交互视图	顺序图	交互、对象、消息、激活
		协作图	协作、交互、协作角色、消息

续表

主要的域	视　图	图	主要概念
模型管理	模型管理视图	类图	包、子系统、模型
可扩展性	所有	所有	约束、构造型、标记值

从历年考题来看，主要集中在类图、用例图、序列图及状态转换图上，尤其是类图、类的属性和方法的识别以及类间的各种关系需要重点掌握。

1. 类图

类图展现了一组对象、接口、协作及其之间的关系。在面向对象系统的建模中所建立的最常见的图就是类图(Class Diagram)。

2. 用例图

用例图(Use Case Diagram)展现了一组用例、参与者(Actor)以及两者之间的关系。用例图通常包括用例、参与者、扩展关系、包含关系。

3. 构件图

构件图(Component Diagram)展现了一组构件之间的组织和依赖关系。构件图专注于系统的静态实现视图。它与类图相关，通常把构件映射为一个或多个类、接口或协作。

4. 部署图

部署图(Deployment Diagram)展现了运行处理结点以及其中的构件的配置。部署图给出了体系结构的静态实施视图。它与构件图相关，通常一个结点包含一个或多个构件。

5. 状态图

状态图(State Diagram)展现了一个状态机，它由状态、转换、事件和活动组成。状态图关注系统的动态视图，它对接口、类和协作的行为建模尤为重要，它强调对象行为的事件顺序。状态图通常包含简单状态和组合状态、转换(事件和动作)。

6. 活动图

活动图(Activity Diagram)是一种特殊的状态图，它展现了在系统内从一个活动到另一个活动的流程。活动图专注于系统的动态视图，它对于系统的功能建模特别重要，并强调对象间的控制流程。活动图一般包括活动状态和动作状态、转换和对象。当对一个系统的动态方面进行建模时，通常有两种使用活动图的方式：对工作流建模；对操作建模。

7. 交互图

顺序图(或称序列图，Sequence Diagram)和协作图(Collaboration Diagram)均被称为交互图(Interactive Diagram)，它们用于对系统的动态方面进行建模。一张交互图显示的是一个交互，由一组对象及其之间的关系组成，包含它们之间可能传递的消息。

顺序图是强调消息时间序列的交互图，协作图则是强调接收和发送消息的对象的结构组织的交互图。

交互图用于对一个系统的动态方面建模。在大多数情况下，它包括对类、接口、构件和结点的具体的或原型化的实例及其之间传递的消息进行建模。交互图可以单独使用，用于可视化、详述、构造和文档化一个特定的对象群体的动态方面，也可以用来对一个用例的特定控制流进行建模。

真题链接

【例 10-9】UML 图中，一张交互图显示一个交互，由一组对象及其之间的关系组成，包含它们之间可能传递的消息。__(45)__不是交互图。(2014 年 11 月真题 45)

(45) A. 序列图　　B. 对象图　　C. 通信图　　D. 时序图

【解析】对象图(Object Diagram)显示了一组对象和它们之间的关系。使用对象图来说明数据结构，类图中的类或组件等的实例的静态快照。

【答案】B

【例 10-10】对一个复杂用例中的业务处理流程进行进一步建模的最佳工具是 UML __(41)__。(2014 年 5 月真题 41)

(41) A. 状态图　　B. 顺序图　　C. 类图　　D. 活动图

【解析】活动图是 UML 用于对系统的动态行为建模的一种常用工具，它描述活动的顺序，展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程。

【答案】D

【例 10-11】如图 10-2 所示的 UML 序列图中，__(42)__表示返回消息，Account 类必须实现的方法有__(43)__。(2014 年 5 月真题 42～43)

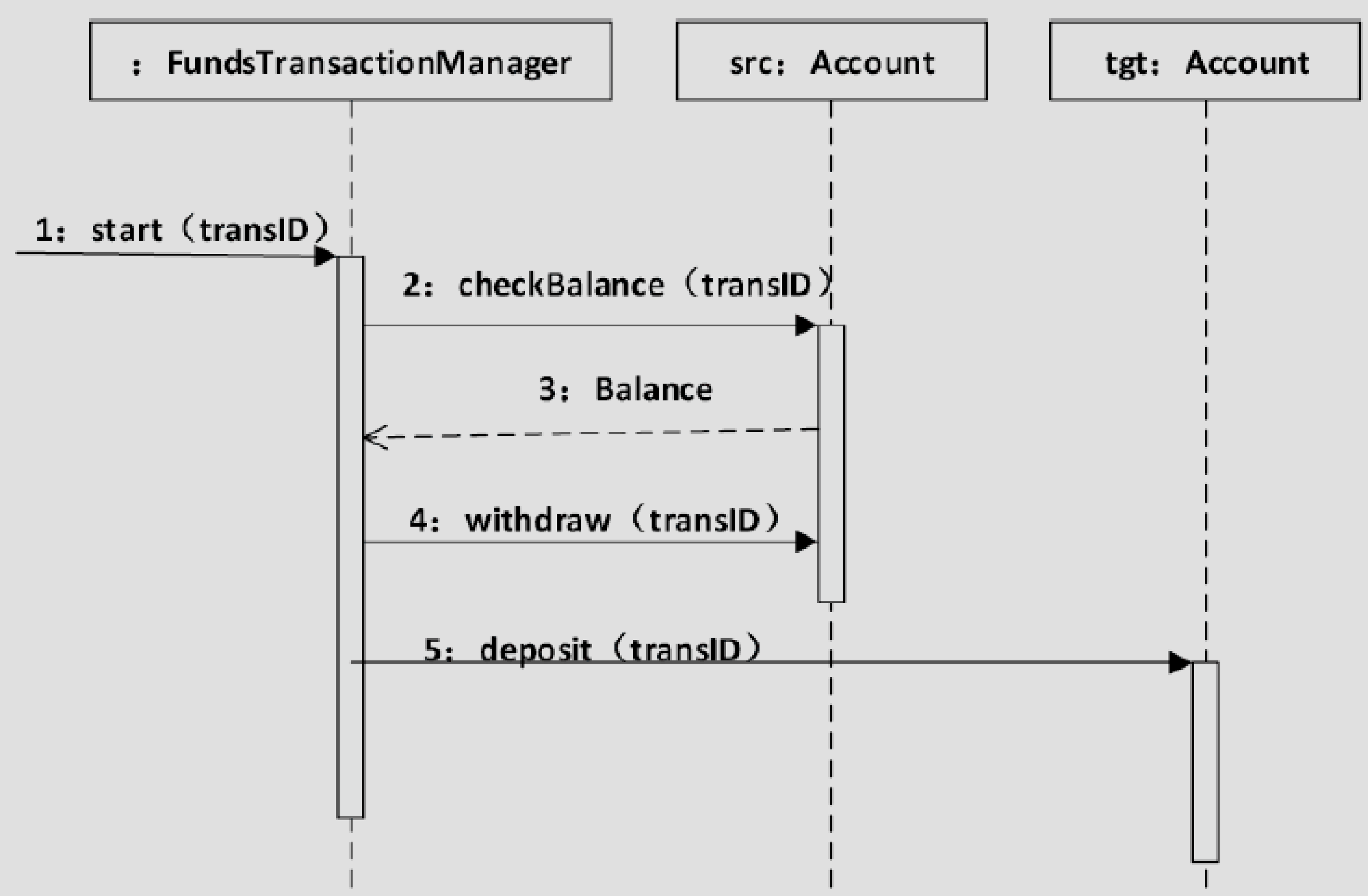


图 10-2　UML 序列图

(42) A. tansID　　B. balance　　C. withdraw　　D. deposit

(43) A. start0　　B. checkBalance()和 withdraw()

C. deposit0　　D. checkBalance()、withdraw()和 deposit()

【解析】在 URL 序列图中，调用消息用带实心箭头的实线表示，返回消息用带箭头的虚线表示，图中的编号为 1、2、3、5 的消息均为调用消息，只有编号为 3 的消息为返回消息。由题目可知，FundTransactionManager 要调用类 Account 的 checkBalance()、withdraw()和 deposit()方法，类 Account 需要实现这 3 个方法。

【答案】(42)B；(43)D

【例 10-12】(2014 年 11 月下午试题三)

阅读以下说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某公司欲开发一个管理选民信息的软件系统。系统的基本需求描述如下：

(1) 每个人(Person)可以是一个合法选民(Eligible)或者无效的选民(Ineligible)。

(2) 每个合法选民必须通过该系统对其投票所在区域(即选区，Riding)进行注册(Registration)。每个合法选民仅能注册一个选区。

(3) 选民所属选区由其居住地址(Address)决定。假设每个人只有一个地址，地址可以是镇(Town)或者城市(City)。

(4) 某些选区可能包含多个镇；而某些较大的城市也可能包含多个选区。

现采用面向对象方法对该系统进行分析与设计，得到如图 10-3 所示的初始类图。

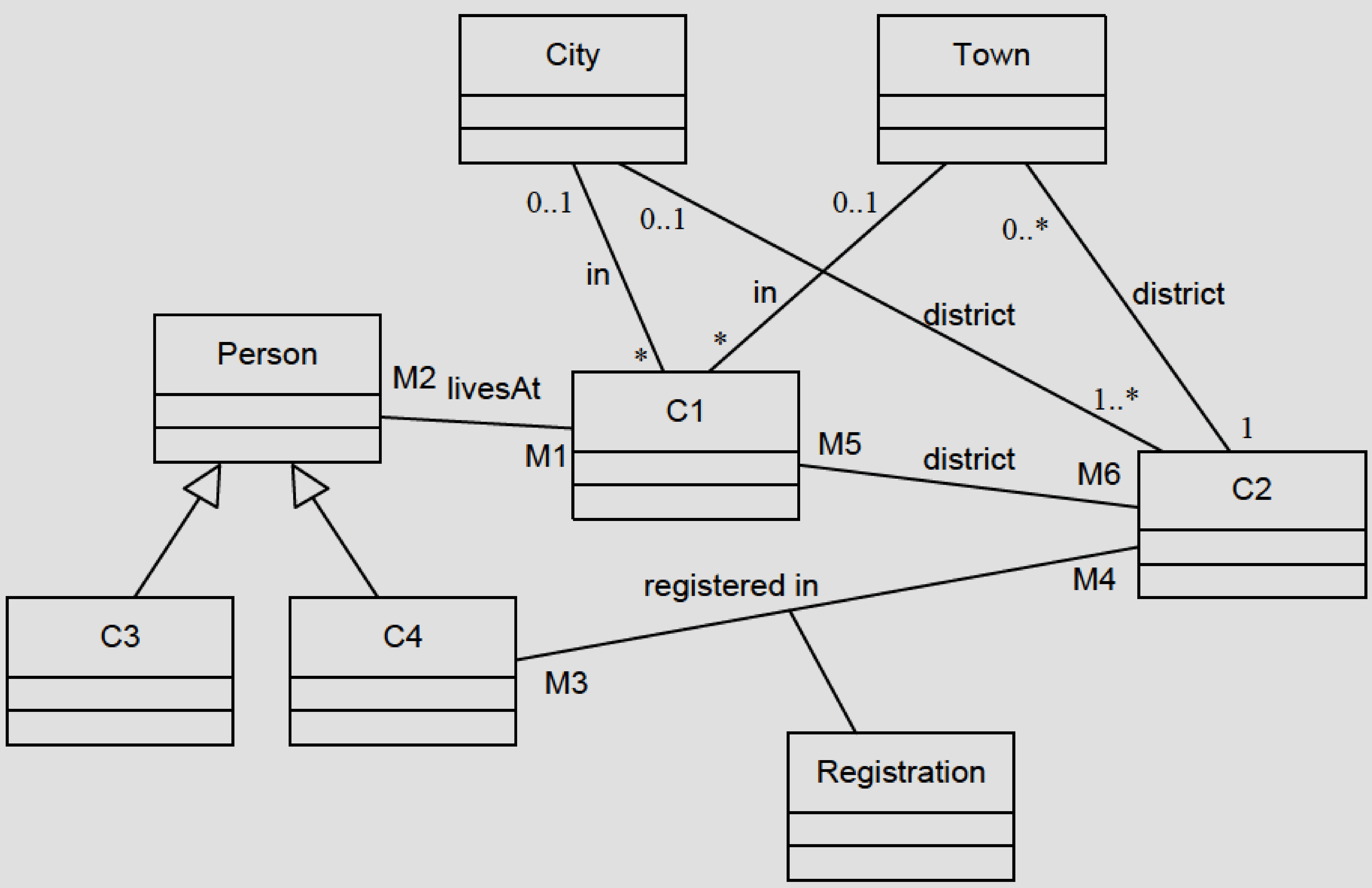


图 10-3　类图

【问题 1】

根据说明中的描述，给出图 10-3 中 C1～C4 所对应的类名(类名使用说明中给出的英文词汇)。

【问题 2】

根据说明中的描述，给出图 10-3 中 M1～M6 处的多重度。

【问题 3】

现对该系统提出了以下新需求：

(1) 某些人拥有在多个选区投票的权利，因此需要注册多个选区；

(2) 对于满足(1)的选民，需要划定其“主要居住地”，以确定他们应该在哪个选区进行投票。

为了满足上述需求，需要对图 10-3 所示的类图进行哪些修改？请用 100 字以内文字说明。

【解析】

【问题 1】

由“每个人可以是一个合法选民或者无效选民”可知 C3 和 C4 是这两者中的一个，由 C4 和 C2 关联可知，C4 为合法选民(Eligible)，则 C3 为无效选民(Ineligible)。由 City 和 Town 共同指向 C1 以及描述“选民所属选区由其居住地址(Address)决定。假设每个人只有一个地址，地址可以是镇(Town)或者城市(City)”可知，C1 应该为 Address。由描述“每个合法选民必须通过该系统对其投票所在区域(即选区，Riding)进行注册(Registration)”以及 C4 指向 C2 和 City、Town 同时指向 C2 可知，C2 为 Riding，即选区。

【问题 2】

UML 中关联的多重度是指一个类的实例能够与另一个类的多少个实例相关联。具体的取值意义有：

0..1：0 个或 1 个

1：只能 1 个

0..*：0 个或多个

*：0 个或多个

1..*：1 个或多个

由描述“每个人只有一个地址”可知，M1 为 1；一个地址可以有 0 个或多个人，因此 M2 为*；一个选区可以有 0 个或多个选民，一个选民只在一个选区投票，因此，M3 为*，M4 为 1；由“某些选区可能包含多个镇”可知每个选区可包含 0 个或多个地址，M5 应为*，每个选民只有一个地址，每个地址属于一个选区，因此 M6 为 1。

【问题 3】

若对系统提出新的要求：①某些人拥有在多个选区投票的权利，因此需要注册多个选区；②对于满足(1)的选民，需要划定其“主要居住地”，以确定他们应该在哪个选区进行投票。则将 M1 与 M4 由 1 修改为 1..*。

【答案】

【问题 1】

C1：Address　C2：Riding　C3：Ineligible　C4：Eligible

【问题 2】

M1: 1，M2:*，M3:*，M4：1，M5:*，M6:1

【问题 3】

将 M1 与 M4 由 1,修改为 1..*

【例 10-13】(2013 年 5 月下午试题三)

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸对应栏内。

【说明】某城市拟开发一个基于 Web 的城市黄页，公开发布该城市重要的组织或机构(以下统称为客户)的基本信息，方便城市生活。该系统的主要功能描述如下。

(1) 搜索信息：任何使用 Internet 的网络用户都可以搜索发布在城市黄页中的信息，例如客户的名称、地址、联系电话等。

(2) 认证：客户若想在城市黄页上发布信息，需通过系统的认证。认证成功后，该客户成为系统授权用户。

(3) 更新信息：授权用户登录系统后，可以更改自己在城市黄页中的相关信息，例如变更联系电话等。

(4) 删除客户：对于拒绝继续在城市黄页上发布信息的客户，有系统管理员删除该客户的相关信息。

系统采用面向对象方法进行开发，在开发过程中认定出如表 10-2 所示的类。系统的用例图和类图分别如图 10-4 和图 10-5 所示。

表 10-2　类列表

类　名	说　明
InternetClient	网络用户
CustomerList	客户集，维护城市黄页上的所有客户信息
Customer	客户信息，记录单个客户的信息
RegisteredClient	授权用户
Administrator	系统管理员

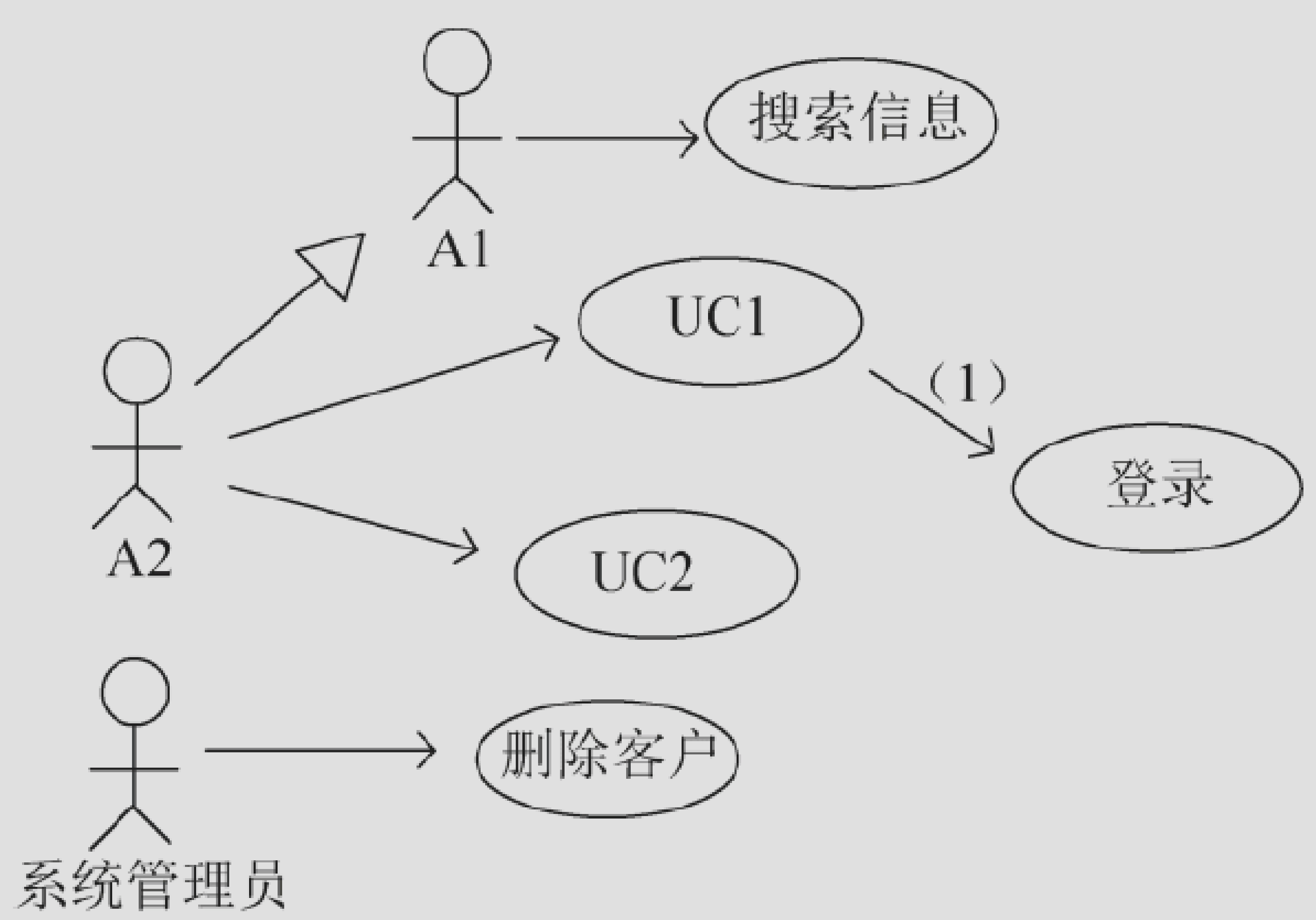


图 10-4　系统用例图

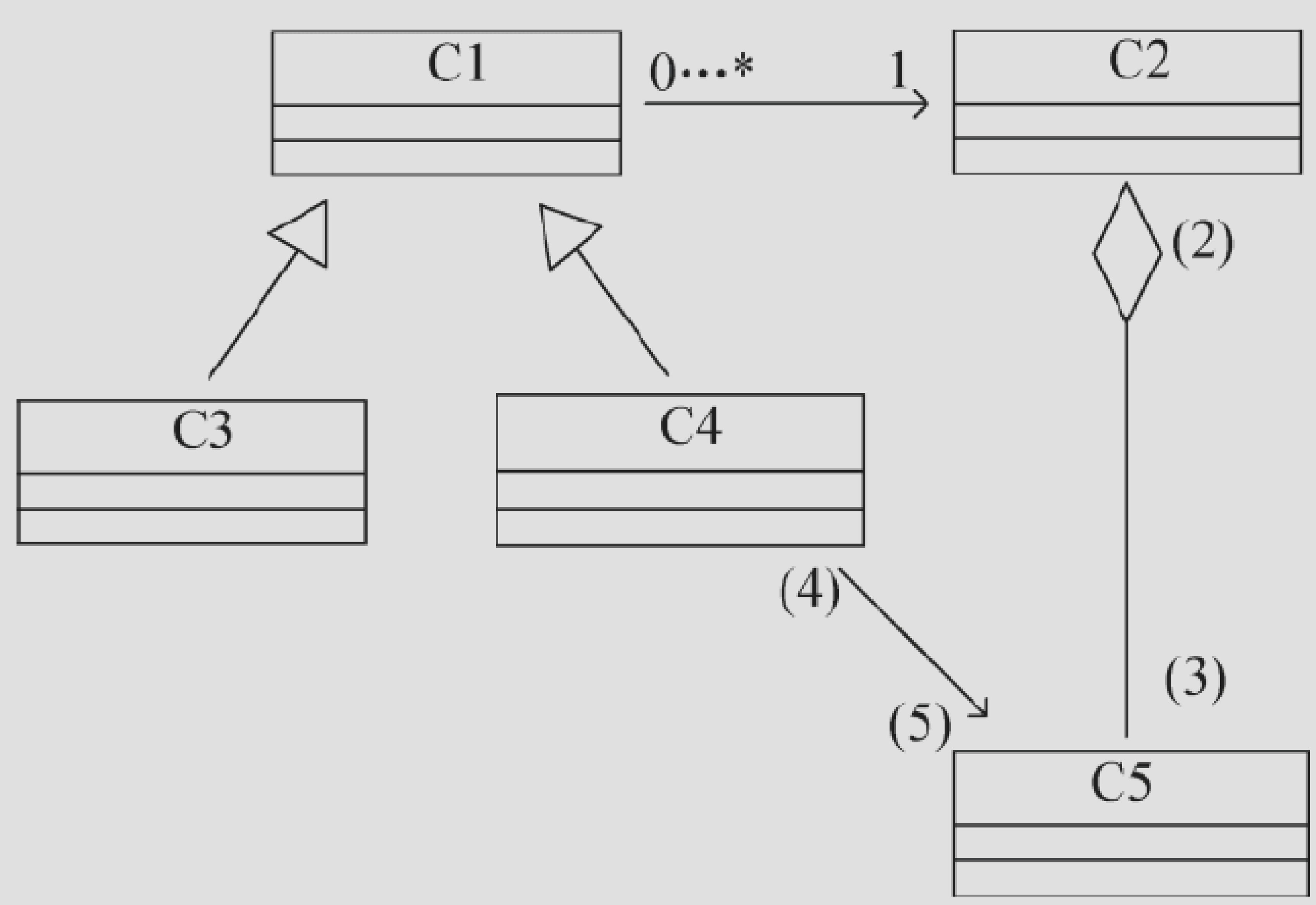


图 10-5　系统类图

【问题 1】

根据说明中的描述，给出图 10-4 中 A1 和 A2 处所对应的参与者，UC1 和 UC2 所对应的用例以及(1)

处的关系。

【问题 2】

根据说明中的描述，给出图 10-5 中 C1～C5 所对应的类名(表 10-2 中给出的类名)和(2)～(5)处所对应的多重度。

【问题 3】

认定类是面向对象分析中非常关键的一个步骤。一般首先从问题域中得到候选类集合，再根据相应的原则从该集合中删除不作为类的，剩余的就是从问题域中认定出来的类。简要说明选择候选类的原则，以及对候选类集合进行删除的原则。

【解析】

本题考查面向对象分析中的类图、用例图。用例图描述了一组用例、参与者及他们之间的关系。包括以下几个部分：用例(Case)、参与者(Actor)。用例视图中的参与者与系统外部的一个实体以某种方式参与了用例的执行过程；用例是一个叙述型文档，用来描述参与使用系统、完成某个事情时发生的顺序。

【问题 1】用例图中，A1 可以搜索信息，A2 由 A1 派生且 A2 参与了两个用例，根据题中的说明 (1)和(2)，可知 A1 为网络用户，A2 为授权用户；由用例 UC1 和登录用例之间存在关系，可知 UC1 为认证用例，因为用户登录必须先认证，所以登录用例是认证的扩展，所以他们之间的关系为 extend。对于授权用户还可以更新信息，故 UC2 为更新信息用例。

【问题 2】本问题考察类图的层次结构和多重度。图 10-4 中更有两个非常明显的继承结构，即 C3 和 C4 继承于 C1，且 C1 与 C2 是多比一的关系，根据说明(1)中任何网络用户都可以搜索客户信息，即 C1 为网络用户，C2 为客户信息，由此很明显地得出 C3 和 C4 在授权用户和系统管理员中选取。根据 C2 和 C5 之间存在聚合关系，且 C2 为客户信息，可以推断 C5 为客户集。再由 C4 和 C5 之间的关联关系，且 C5 为客户集，能对客户集进行批量操作的用户 C4 显然就是系统管理员，由此得出 C3 为授权用户。由此(2)～(5)的多重度就显而易见，(2)为 1，(3)为 0…*，(4)为 1，(5)为 0…*。

【问题 3】候选类的选择运用了良性依赖原则“不会在实际中造成危害的依赖关系，都是良性依赖”。通过分析不难发现，本原则的核心思想是“务实”，很好地揭示了极限编程(Extreme Programming)中“简单设计”和“重构”的理论基础。本原则可以帮助我们抵御“面向对象设计五大原则”以及设计模式的诱惑，以免陷入过度设计(Over-engineering)的尴尬境地，带来不必要的复杂候选类的删除。

候选类的删除运用了接口隔离原则(ISP)：不应该强迫客户依赖于它们不用的方法。接口属于客户，不属于它所在的类层次结构。这个说得很明白了，再通俗点说就是不要强迫客户使用他们不用的方法，如果强迫用户使用他们不使用的方法，那么这些客户就会面临由于这些不使用的方法的改变所带来的改变。

【答案】

【问题 1】A1：网络用户　A2：授权用户　UC1：认证　UC2：更新信息　(1) extend

【问题 2】

C1：InternetClient　C2：Customer　C3：RegisteredClient　C4：Administrator

C5：CustomerList

(2)1　(3)0...*　(4)1　(5)0...*

【问题 3】候选类的选择运用了良性依赖原则“不会在实际中造成危害的依赖关系，都是良性依赖”和接口隔离原则(ISP)。

【例 10-14】(2013 年 11 月下午试题三)

阅读以下说明和图，根据要求回答问题 1～问题 3。

【说明】

某航空公司会员积分系统(CFrequentFlyer)的主要功能描述如下：

乘客只要办理该航空公司的会员卡，即可成为普卡会员(CBasic)。随着飞行里程数的积累，可以从普卡会员升级到银卡会员(CSilver)或金卡会员(CGold)。非会员(CNonMember)不能累积里程数。

每年年末，系统根据会员在本年度累积的里程数对下一年会员等级进行调整。

普卡会员在一年内累积的里程数若满 25 000 英里但不足 50 000 英里，则自动升级为银卡会员；若累积的里程数在 50 000 英里以上，则自动升级为金卡会员。银卡会员在一年内累积的里程数若在 50 000 英里以上，则自动升级为金卡会员。

若一年内没有达到对应级别要求的里程数，则自动降低会员等级。金卡会员一年内累积的里程数若不足 25 000 英里，则自动降级为普卡会员；若累积的里程数达到 25 000 英里，但是不足 50 000 英里，则自动降级为银卡会员。银卡会员一年内累积的里程数若不足 25 000 英里，则自动降级为普卡会员。

采用面向对象方法对会员积分系统进行分析与设计，得到如图 10-6 所示的状态图和图 10-7 所示的类图。

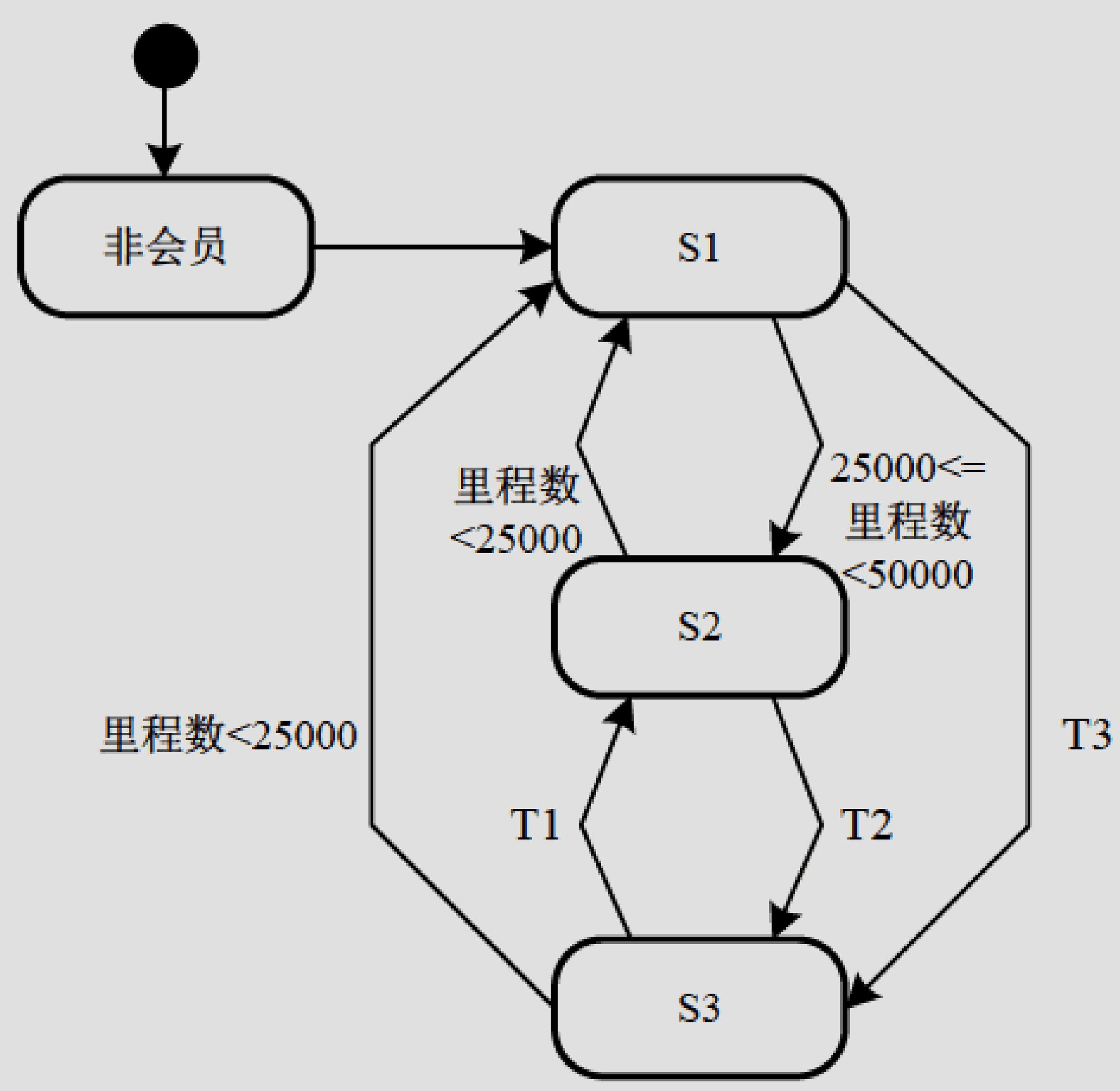


图 10-6　某会员积分系统状态图

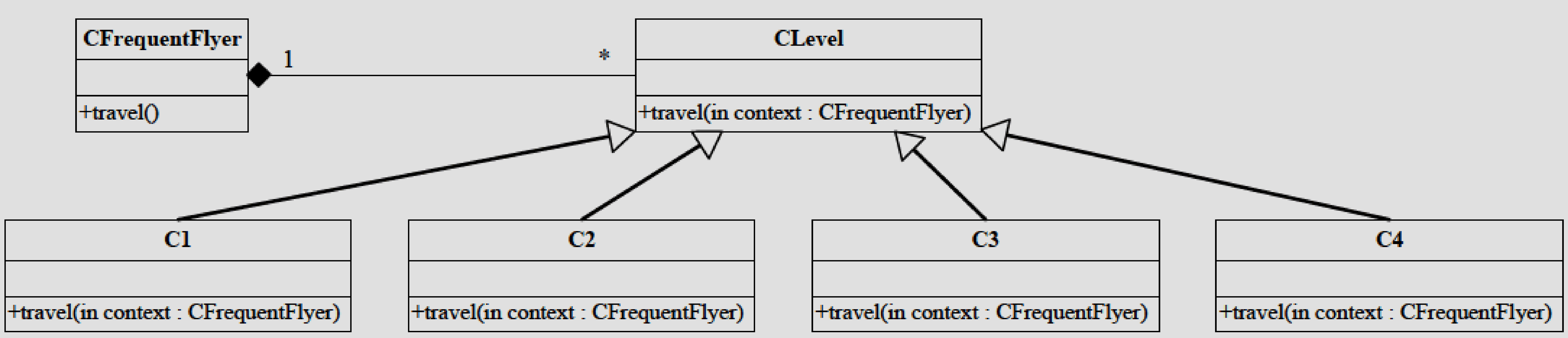


图 10-7　某会员积分系统类图

【问题 1】

根据说明中的描述，给出图 10-6 中 S1～S3 处所对应的状态以及 T1～T3 处所对应的迁移的名称。

【问题 2】

根据说明中的描述，给出图 10-7 中 C1～C4 所对应的类名(类名使用说明中给出的英文词汇)。

【问题 3】

图 10-7 所示的类图中使用了哪种设计模式？在这种设计模式下，类 CFrequemFlver 必须具有的属性是什么？C1～C4 中的 travel 方法应具有什么功能？

本题考查面向对象分析中的类图、用例图。用例图描述了一组用例、参与者及他们之间的关系。包括以下几个部分：用例(Case)、参与者(Actor)。会员积分系统状态图就是一种用例图。用例视图中的参与者与系统外部的一个实体以某种方式参与了用例的执行过程；用例是一个叙述型文档，用来描述参与使用系统、完成某个事情时发生的顺序。

【解析】

【问题 1】图中要求填充 S1、S2、S3 这三个状态以及它们之间的变迁关系。本题中会员有三种状态：普卡、金卡和银卡。普卡会员在一年内累积的里程数若满 25 000 英里但不足 50 000 英里，则自动升级为银卡会员；若累积的里程数在 50 000 英里以上，则自动升级为金卡会员。银卡会员在一年内累积的里程数若在 50 000 英里以上，则自动升级为金卡会员；所以，S1 为普卡会员、S2 为银卡会员、S3 为金卡会员。同样根据上述分析可知，T1、T2 就是 S2 和 S3 之间的转换原则，T3 是 Sl→S2 的转换原则。

【问题 2】本问题考察类图。考察类图的层次结构和多重度。图中更有 4 个非常明显的继承结构，即 C1～C4 继承 CLevel，根据说明可知代表了四类不同的会员模式。

【问题 3】状态模式允许对象在内部状态变化时，变更其行为，并且修改其类。状态模式的类图如图 10-8 所示。

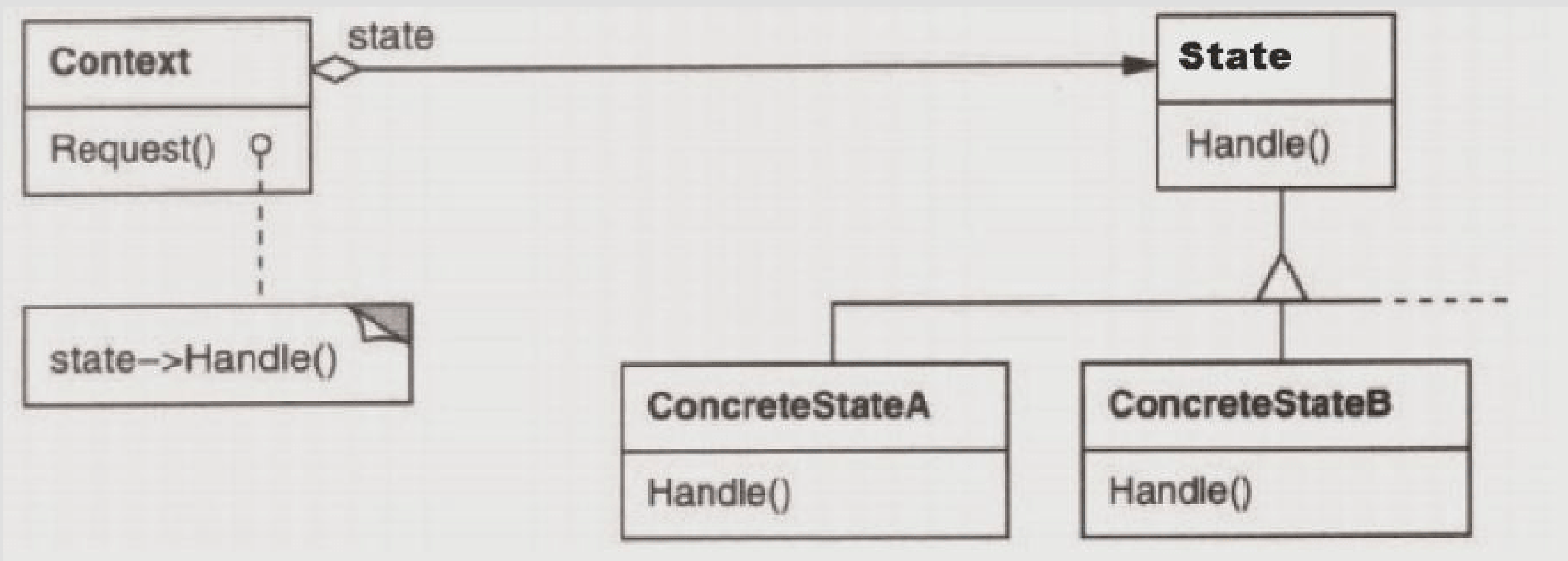


图 10-8　状态模式的类图

状态模式的组成如下：

环境类(Context)：定义客户感兴趣的接口。维护一个 ConcreteState 子类的实例，这个实例定义当前状态。

抽象状态类(State)：定义一个接口以封装与 Context 的一个特定状态相关的行为。

具体状态类(ConcreteState)：每一子类实现一个与 Context 的一个状态相关的行为。

系统类图中的类 CFrequentFlyer 对应图 10-8 中的环境类，因此类 CFrequentFlyer 应该有一个 CLevel 类的对象。

Travel 方法的功能：计算飞行里程数，根据里程数判断是否需要调整会员级别(跳转到不同的状态)。

【答案】

【问题 1】

S1：普卡会员

S2：银卡会员

S3：金卡会员

T1：25000 <= 里程数 < 50000

T2：里程数 >= 50000

T3：里程数 >= 50000

【问题 2】

C1：CNonMember

C2：CBasic

C3：CSilver

C4：CGold

【问题 3】

使用了 State 模式(状态模式)。

类 CFrequentFlyer 必须具有的属性：CLevel 的对象。

Travel 方法的功能：计算飞行里程数，根据里程数判断是否需要调整会员级别(跳转到不同的状态)。

10.6　设 计 模 式

考核说明：本节主要介绍设计模式的要素、创建型设计模式、结构型设计模式和行为设计模式，各种具体的设计模式是上午必考内容，要重点掌握。

10.6.1　设计模式的要素

设计模式一般有如下四个要素。

(1) 模式名称(Pattern Name)。一个助记名，它用一两个词来描述模式的问题、解决方案和效果。设计模式允许在较高的抽象层次上进行设计。基于一个模式词汇表，就可以讨论模式并在编写文档时使用它们。模式名可以帮助人们思考，便于人们与其他人交流设计思想及设计结果。

(2) 问题(Problem)。描述了应该在何时使用模式。它解释了设计问题和问题存在的前因后果，可能描述了特定的设计问题，如怎样用对象表示算法等；也可能描述了导致不灵活设计的类或对象结构。

(3) 解决方案(Solution)。描述了设计的组成成分、它们之间的相互关系及各自的职责和协作方式。因为模式就像一个模板，可应用于多种不同场合，所以解决方案并不描述一个特定而具体的设计或实现，而是提供设计问题的抽象描述和怎样用一个具有一般意义的元素组合(类或对象组合)来解决这个问题。

(4) 效果(Consequences)。描述了模式应用的效果及使用模式应权衡的问题。尽管描述设计决策时，并不总提到模式效果，但它们对于评价设计选择和理解使用模式的代价及好处具有重要意义。

10.6.2 创建型设计模式

创建型设计模式抽象了实例化过程，可以帮助一个系统独立于创建、组合和表示它的那些对象。一个类创建型模式使用继承改变被实例化的类，而一个对象创建型模式将实例化委托给另一个对象。

创建型模式中有两个不断出现的主旋律。第一，它们都将关于该系统使用哪些具体的类的信息封装起来。第二，它们隐藏了这些类的实例是如何被创建和放在一起的。整个系统关于这些对象所知道的是由抽象类所定义的接口。因此，创建型模式在什么被创建，谁创建它，它是怎样被创建的，以及何时创建这些方面给予了很大的灵活性。创建型设计模式允许用结构和功能差别很大的“产品”对象配置一个系统。配置可以是静态的(即在编译时指定)，也可以是动态的(在运行时)。

10.6.3 结构型设计模式

结构型设计模式涉及如何组合类和对象以获得更大的结构。结构型设计模式采用继承机制来组合接口或实现。结构型对象模式不是对接口和实现进行组合，而是描述了如何对一些对象进行组合，从而实现新功能的一些方法。

Composite(组合)模式是结构型对象模式的一个实例。它描述了如何构造一个类层次式结构，这一结构由两种类型的对象所对应的类构成。

Proxy(代理)模式通过提供与对象相同的接口来控制对这个对象的访问。

Flyweight(享元)模式为共享对象定义了一个结构。至少有两个原因要求对象共享：效率和一致性。Flyweight 模式的对象共享机制主要强调对象的空间效率。使用很多对象的应用必须考虑每一个对象的开销。

Facade(外观)模式描述了如何用单个对象表示整个子系统。模式中的 Facade 用来表示一组对象，Facade 的职责是将消息转发给它所表示的对象。

Bridge(桥接)模式将对象的抽象和其实现分离，从而可以独立地改变它们。

Decorator(装饰器)模式描述了如何动态地为对象添加职责。这一模式采用递归方式组合对象，允许添加任意多的对象职责。

真题链接

【例 10-15】图 10-9 所示为 (46) 设计模式，适用于 (47) 。(2014 年 11 月真题 46～47)

(46) A. 抽象工厂(Abstract Factory)　　B. 生成器(Builder)

C. 工厂方法(Factory Method)　　D. 原型(Prototype)

(47) A. 一个系统要由多个产品系列中的一个来配置时

B. 当一个类希望由它的子类来指定它所创建的对象时

C. 当创建复杂对象的算法应该独立于该对象的组成部分及其装配方式时

D. 当一个系统应该独立于它的产品创建、构成和表示时

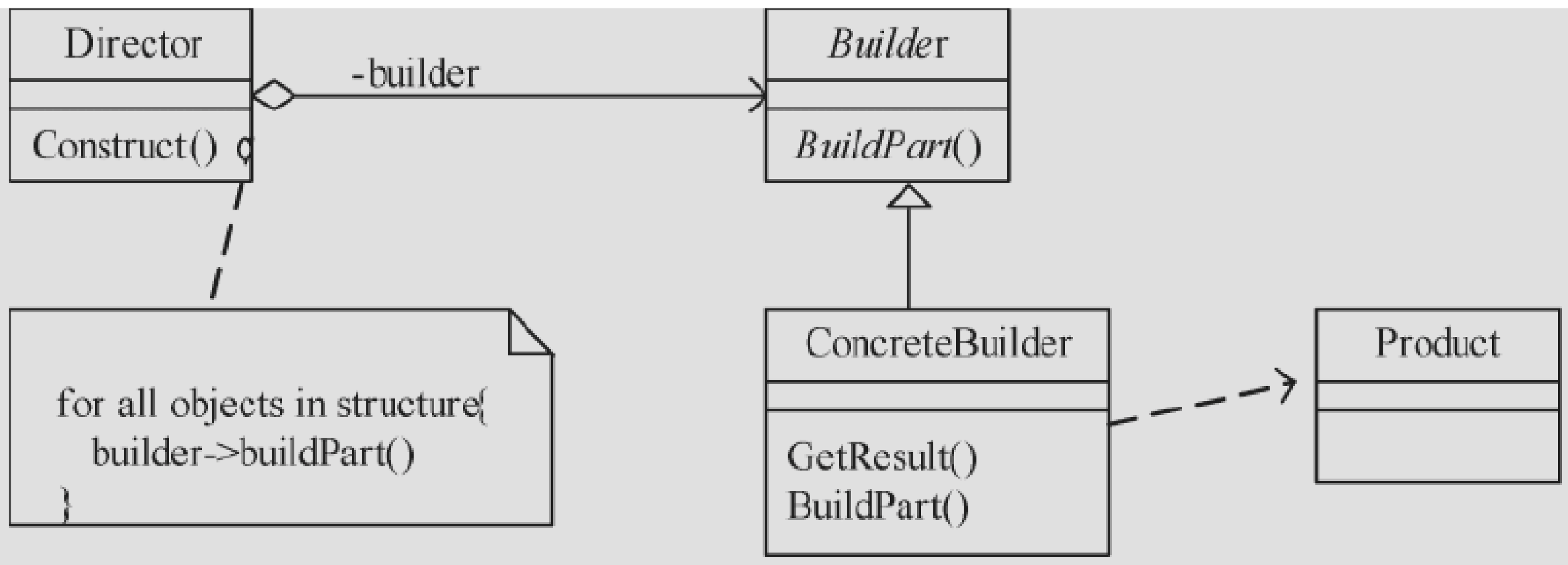


图 10-9　某设计模式类图

【解析】生成器又称建造模式，是一种对象构建模式。它可以将复杂对象的建造过程抽象出来(抽象类别)，使这个抽象过程的不同实现方法可以构造出不同表现(属性)的对象。该模式通常包含 Builder、ConcreteBuilder, Director 和 Product 四部分。

【答案】(46)B；(47)C

【例 10-16】图 10-10 所示为 (44) 设计模式，适用于 (45) 。(2014 年 5 月真题 44～45)

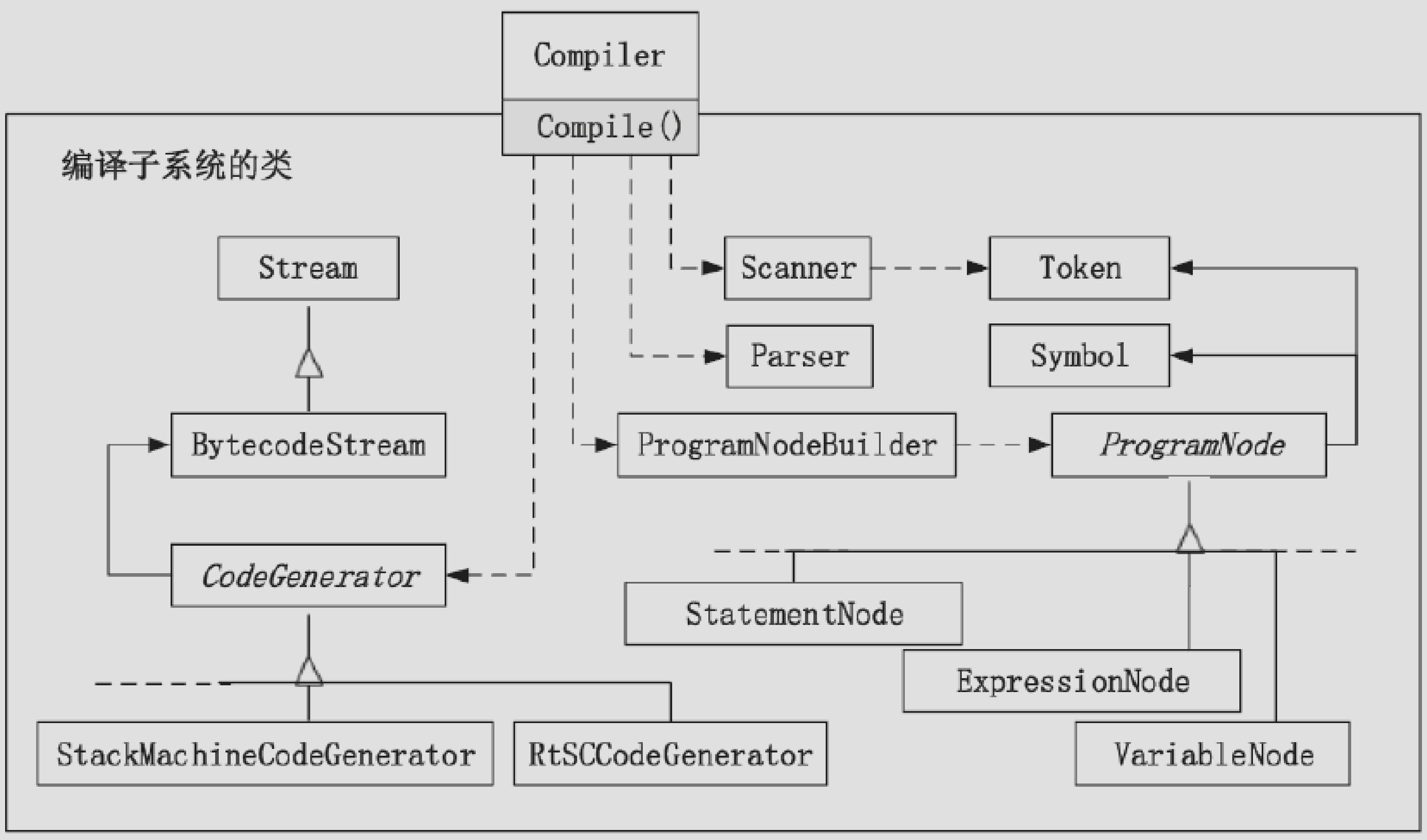


图 10-10　某设计模式类图

(44) A. 适配器(Adapter) B. 责任链(Chain of Responsibility) C. 外观(Facade) D. 桥接(Bridge)

(45) A. 有多个对象可以处理一个请求，在运行时刻自动确定由哪个对象处理

B. 想使用一个已经存在的类，而其接口不符合要求

C. 类的抽象和其实现之间不希望有一个固定的绑定关系

D. 需要为一个复杂子系统提供一个简单接口

【解析】外观设计模式要求一个子系统的外部与其内部的通信必须通过一个统一的对象进行交互。外观设计模式提供一个高层次的接口使得子系统更易于使用。

外观模式适用情况如下：

(1) 为复杂的子系统提供一个简单的接口；

(2) 客户程序与抽象类的实现部分之间存在着很大的依赖性；

(3) 构建一个层次结构的子系统时，适用外观模式定义子系统中每层的入口点。

【答案】(44)C；(45)D

10.6.4 行为设计模式

行为设计模式涉及算法和对象间职责的分配。行为模式不仅描述对象或类的模式，还描述它们之间的通信模式。这些模式刻画了在运行时难以跟踪的复杂的控制流。它们将用户的注意力从控制流转移到对象间的联系方式上来。

行为类模式使用继承机制在类间分派行为，主要有 TemplateMethod 和 Interpreter 两种模式。

行为对象模式使用对象复合而不是继承。一些行为对象模式描述了一组对等的对象怎样相互协作以完成其中任意一个对象都无法单独完成的任务。

行为设计模式包括责任链(Chain of Responsibility)、命令(Command)、解释器(Interpreter)、迭代器(Iterator)、中介者(Mediator)、备忘录(Memento)、观察者(Observer)、状态(State)、策略(Strategy)、模板方法(Template Method)、访问者(Visitor)等模式。其中，Obsever(观察者)模式定义并保持对象间的依赖关系。典型的 Observer 的例子就是 Smalltalk 中的模型/视图/控制器，其中一旦模型的状态发生变化，模型的所有视图都会得到通知。

其他的行为对象模式常将行为封装在一个对象中并将请求指派给它。

真题链接

【例 10-17】一组对象以定义良好但是复杂的方式进行通信，产生的相互依赖关系结构混乱且难以理解。采用 (43) 模式，用一个中介对象来封装一系列的对象交互，从而使各对象不需要显式地相互引用，使其耦合松散，而且可以独立地改变它们之间的交互。此模式与 (44) 模式是相互竞争的模式，主要差别是：前者的中介对象封装了其他对象间的通信，而后者通过引入其他对象来分布通信。(2014 年 11 月真题 43、44)

(43) A. 解释器(Interpreter) B. 策略(Strategy) C. 中介者(Mediator) D. 观察者(Observer)

(44) A. 解释器(Interpreter) B. 策略(Strategy) C. 中介者(Mediator) D. 观察者(Observer)

【解析】中介者模式包装了一系列对象相互作用的方式，使得这些对象不必互相明显引用，从而使它们可以较松散地耦合。当这些对象中的某些对象之间的相互作用发生改变时，不会立即影响到其他的一些对象之间的相互作用，从而保证这些相互作用可以彼此独立地变化。在中介者模式中，所有的成员对象者可以协调工作，但是又不直接相互管理。这些对象都与一个处于中心地位的中介者对象发生紧密的关系，由这个中介者对象进行协调工作。这个协调者对象称为中介者(Mediator)，而中介者所协调的成员对象称为同事(Colleague)对象。

在观察者模式中，一个目标对象管理所有相依于它的观察者对象，并且在它本身的状态改变时主动发出通知。这通常通过调用各观察者所提供的方法来实现。此种模式通常被用来实现事件处理系统。

【答案】(41)C；(44)D

10.7 C++程序设计

考核说明：本节主要介绍 C++程序设计中的要点，要掌握类的继承、派生和虚函数的使用，这是下午必考内容。

10.7.1　类和对象

1. 类的定义格式

类的定义格式如下：

```
class 类名{
private:
    //私有数据和函数
public:
    //公有数据和函数
protected:
    //保护数据和函数
};
```

其中，private、public、protected 称为访问权限控制关键字，其作用是限制“可见性”。在类的定义中，以上三种关键字出现的次数和先后次序都没有限制，默认被认为是私有成员。

成员函数通常在类的说明之后定义，其格式如下：

```
返回值类型类名::函数名(参数表)
{
    //函数体
}
```

其中，运算符“::”称为作用域解析运算符，用于指出该函数是属于哪一个类的成员函数。当然也可以在类的定义中直接定义函数。

2. 对象

对象即类的实例(Instance)。创建类的对象有两种常用方法。

第一种是直接对象，CGoods Car；这个定义创建了 CGoods 类的一个对象 Car，同时为它分配了属于自己的存储块，用来存放数据和对这些数据实施操作的成员函数(代码)。与变量定义一样，一个对象只在定义它的域中有效。通过“对象名.成员名”方式访问对象的成员。

第二种是动态创建类的对象，当然变量也同样可动态创建。所谓动态是指在程序运行时建立对象，而前一种是在编译时(程序运行前)建立。通过对象指针访问对象成员。

10.7.2　构造函数和析构函数

1. 构造函数

构造函数是特殊的成员函数，其特征如下。

(1) 成员函数的一种，名称与类名相同，可以有参数，不能有返回值(void 也不行)。

(2) 作用是对对象进行初始化，如给成员变量赋初值。

(3) 如果定义类时没写构造函数，则编译器生成一个默认的无参数的构造函数。默认

构造函数无参数，什么也不做。

(4) 如果定义了构造函数，则编译器不生成默认的无参数的构造函数。

(5) 对象生成时构造函数自动被调用。

(6) 一个类可以有多个构造函数，它们的参数个数或类型不同，构成函数重载。

构造函数一般声明为 public，当然有时为了特殊需要也可定义为 private。

2. 复制构造函数

同一个类的对象在内存中有完全相同的结构，如果作为一个整体进行复制是完全可行的。这个复制过程只需要复制数据成员，而函数成员是共用的(只有一份副本)。

在建立对象时可用同一类的另一个对象来初始化该对象，这时所用的构造函数称为复制初始化构造函数(Copy Constructor)，形如 X::X(X&)，只有一个参数——同类(Class)的对象，采用的是引用的方式。不允许有形如 X::X(X)的构造函数，如果把一个真实的类对象作为参数传递到复制构造函数，会引起无穷递归。如果没有定义，那么编译器生成默认复制构造函数。如果定义了自己的复制构造函数，则不存在默认的复制构造函数。

3. 析构函数

当定义一个对象时，C++自动调用构造函数建立该对象并进行初始化；那么当一个对象的生命周期结束时，C++也会自动调用一个函数注销该对象并进行善后工作，这个特殊的成员函数即析构函数(Destructor)。

(1) 析构函数名与类名相同，但要在前面加上字符“~”，如~CGoods()。

(2) 析构函数无函数返回类型，与构造函数在这方面是一样的。但析构函数不带任何参数。

(3) 一个类有且只有一个析构函数，这与构造函数不同。析构函数可以缺省。

(4) 对象注销时，系统自动调用析构函数。

4. 成员对象

有成员对象的类叫封闭(Enclosing)类。对成员对象初始化，必须调用该成员对象的构造函数来实现。

先调用所有对象成员的构造函数，然后才调用封闭类的构造函数。对象成员的构造函数调用次序和对象成员在类中的说明次序一致，与它们在成员初始化列表中出现的次序无关。当封闭类的对象消亡时，先调用封闭类的析构函数，然后再调用成员对象的析构函数，次序和构造函数的调用次序相反。

10.7.3 继承与派生

1. 基本概念

继承(Inheritance)机制是面向对象程序设计使代码可以复用的最重要手段，它允许程序员在保持原有类特性的基础上，调整部分成员的特性，也可以增加一些新成员。

通过继承，能够以已有的类为基础定义新的类，使新的类具有已有类的全部特点和功

能，新的类还能添加自己的特点和功能，或修改老的类的特点和功能。已有的类(被继承的类)称为基类或父类，新的类(继承的类)称为派生类或子类。

2. 访问控制

C++提供了三种继承方式，也是用 public、protected、private 三个关键字标识，一般采用公有继承 public。三种继承方式的具体意义如表 10-3 所示。

表 10-3　三种继承方式的具体意义

派生方式	基类中的访问限定	在派生类中对基类成员的访问限定	在派生类对象外访问派生类对象的基类成员
公有派生 public	public	public	可直接访问
	protected	protected	不可直接访问
	private	不可访问	不可直接访问
保护派生 protected	public	protected	不可直接访问
	protected	protected	不可直接访问
	private	不可访问	不可直接访问
私有派生 private	public	private	不可直接访问
	protected	private	不可直接访问
	private	不可访问	不可直接访问

3. 赋值兼容规则

在需要基类对象的地方可以使用公有派生类来替代，派生类对象能自动地当作其基类对象来使用，但基类对象不能当作其派生类对象来使用。这正是体现了“派生类对象是一个基类对象”。

具体使用情况如下。

- 派生类的对象可以赋值给基类对象：b = d。
- 派生类对象可以初始化基类引用：base &br = d。
- 派生类对象的地址可以赋值给基类指针：base *pb = &d。

4. 重置(覆盖)

派生类可以定义一个和基类成员同名的成员，这叫覆盖。派生类成员将覆盖所有基类的同名成员，默认的情况是引用派生类的成员，若想访问基类同名成员，需要使用域作用符“::”——基类名::数据成员名、基类名::函数成员名(参数表)。

10.7.4　多态

多态性(Polymorphism)同继承性一样是面向对象程序设计的标志性特征，是一个考查重点。

1. 多态的定义

多态是指同样的消息被不同类型的对象接收时导致不同的行为(不同的实现或调用了不同的函数)。所谓消息，是由“类::方法”(功能)和“方法的实参”(消息数据)共同组成的。

产生多态性的原因是：不同的对象，在处理同样的消息时，使用的方法实现(成员函数的函数体)不同。“多态性”是与“类的派生和继承”联系在一起的，是基类中所定义方法的“多态性”，在派生类中新增加的方法是没有多态性的。

2. 多态的分类

(1) 重载多态——成员函数(运算符)重载。

(2) 强制多态——强制类型转换。把一个变量的类型变换成另一种类型，以符合一个函数或者操作的要求。例如，加法运算符在执行浮点数和整数的相加时，首先把整数转换成浮点数，然后再相加。

(3) 包含多态——不同类中，由虚函数来实现。通过派生-继承关系联系在一起的一组类中，该类多态可使不同类中同名成员函数具有不同的行为。

(4) 参数多态——类模板，操作对象的类型不同，与类模板相关。

在 C++中有两种多态性。

- 编译时的多态性：通过函数的重载和运算符的重载来实现。
- 运行时的多态性：是指在程序执行前，无法根据函数名和参数来确定该调用哪一个函数，必须在程序执行过程中，根据执行的具体情况来动态地确定。这种多态性是通过类继承关系和虚函数(Virtual Function)来实现的。

3. 虚函数

虚函数是前面有 virtual 关键字的类的成员函数，定义虚函数的格式如下：

```
virtual 返回类型函数名(参数表);
```

注意：virtual 关键字只用在类定义的函数声明中，写函数体时不用。另外，如果基类中的函数不是虚函数，即没有 virtual 关键字，即使派生类中的函数写了 virtual 也没有用，不能实现多态。

4. 纯虚函数和抽象类

(1) 纯虚函数(Pure Virtual Function)：指被标明为不具体实现的虚拟成员函数。定义纯虚函数的一般格式为：

```
virtual 返回类型函数名(参数表) = 0;
```

定义纯虚函数必须注意以下几点。

- 定义纯虚函数时，不能定义虚函数的实现部分。
- “=0”本质上是将指向函数体的指针定义为 NULL。
- 在派生类中必须有重新定义的纯虚函数的函数体，这样的派生类才能用来定义对象。

(2) 抽象类：包含纯虚函数的类叫抽象类。

抽象类只能作为基类来派生新类使用，不能创建抽象类的对象，可声明一个抽象类的

指针和引用。

在抽象类的成员函数内可以调用纯虚函数，但是在构造函数或析构函数内部不能调用纯虚函数。因为在构造函数或析构函数内部调用纯虚函数采用的是静态联编，即编译时就要生成调用该函数的指令，而纯虚函数是没有代码的，所以这样的调用指令无法生成，因此编译会报错。在普通成员函数内调用纯虚函数，尽管纯虚函数是没有代码的，但是此时是动态联编，编译时不需要生成调用该函数的指令，所以编译可以通过。在运行时才决定到底调用的是抽象类还是派生类的函数，而抽象类不可能生成对象，所以不可能调用抽象类中的纯虚函数。

5. 虚析构函数

只要基类的析构函数是虚函数，那么派生类的析构函数，不论是否使用 virtual 关键字，不论是用户自己定义的还是编译器默认生成的，都自动成为虚函数。

一个类的构造函数会在执行自己的代码之前，依派生顺序自动调用它的所有直接基类的构造函数；一个类的析构函数也会在执行完自己的代码之后，以与构造函数调用次序相反的顺序自动调用其所有直接基类的析构函数。一般来说，一个类如果定义了虚函数，则应该将析构函数也定义成虚函数。

真题链接

【例 10-18】(2014 年 11 月下午试题五)

阅读下列说明和 C++代码，将应填入 <u>(*n*)</u> 处的字句写在答题纸的对应栏内。

【说明】

某灯具厂商欲生产一个灯具遥控器，该遥控器具有 7 个可编程的插槽，每个插槽都有开关按钮，对应着一个不同的灯。利用该遥控器能够统一控制房间中该厂商所有品牌灯具的开关，现采用 Command(命令)模式实现该遥控器的软件部分。Command 模式的类图如图 10-11 所示。

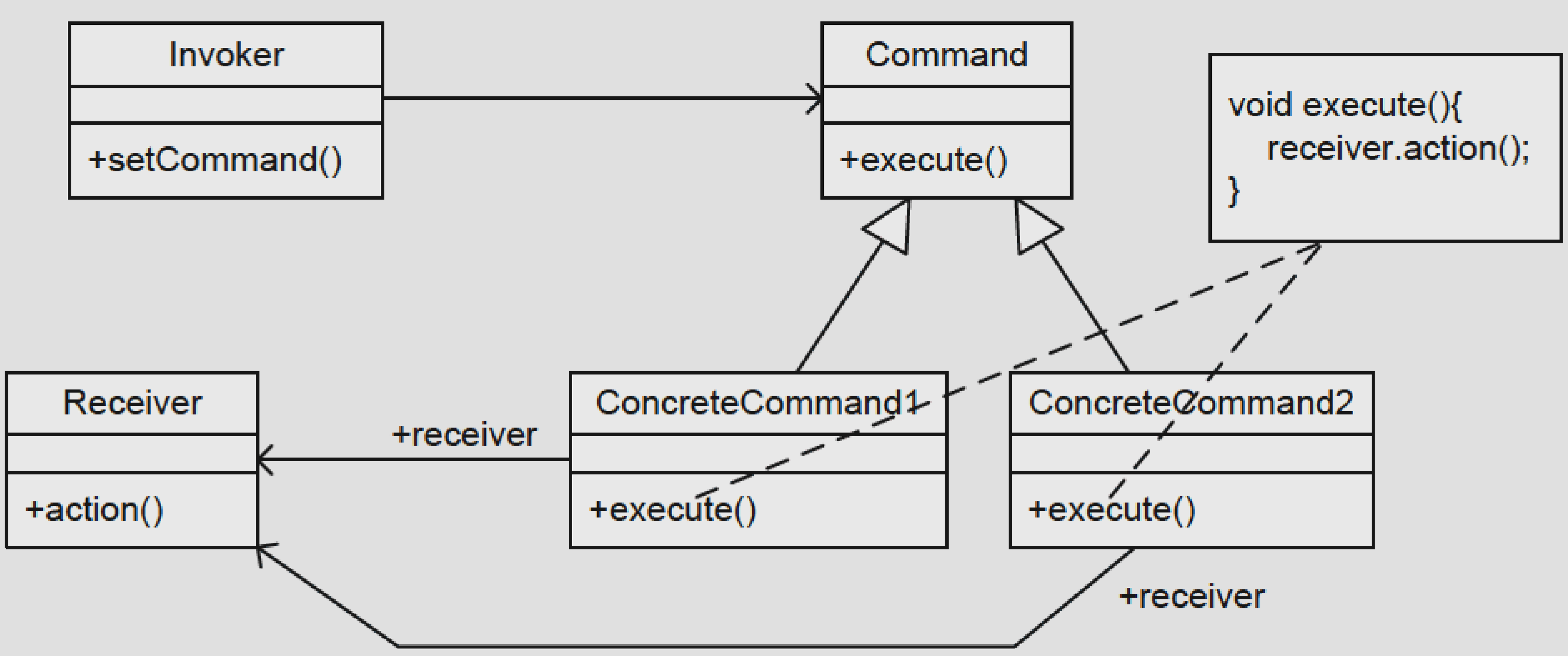


图 10-11　Command 模式类图

【C++代码】

```
class Light{
public:
  Light(string name){/*代码省略*/}
```

```
  void on(){/*代码省略*/}  //开灯
  void off(){/*代码省略*/}  //关灯
}:
class Command{
public:
  __(1)__;
};
class LightOnCommand:public Command{    //开灯命令
private:
  Light* light;
public:
  LightOnCommand(Light*light){this->light=light;}
  Void execute(){__(2)__;}
};
class LightOffCommand:public Command{   //关灯命令
private:
  Light* light;
public:
  LightOffCommand(Light*light){this->light=light;}
  void execute(){__(3)__;}
};
class RemoteControl{                    //遥控器
private:
   Command* onCommands[7];
   Command* offCommands[7];
public:
   RemoteControl() { /*代码省略*/ }
   void setCommand(int slot Command* onCommand, Command* offCommand) {
      __(4)__=onCommand;
      __(5)__=offCommand;
   }
   void onButtonWasPushed(int slot) { __(6)__ :)
   void offButtonWasPushed(int slot) {__(7)__ : }
);
int main() {
   RemoteControl * remoteControl = new RemoteControl ();
   Light* livingRoomLight=new Light("Living Room");
   Light* kitchen Light= new Light("kitchen");
   LightOnCommand* IivingRoomLightOn= new LightOnCommand(livingRoomLight);
   LightOffCommand* livingRoomLightOff=new LightOffCommand(livingRoomLight);
   LightOnCommand* kitchenLightOn=new LightOnCommand(kitchenLight);
   LightOffCommand* kitchenLightOff= new LightOffCommand(kitchenLight);
   remoteControl->setCommand(0, livingRoomLightOn, livingRoomLightOff);
   remoteControl- > setCommand(1, kitchenLightOn, kitchenLightOff);
   remoteControl- >onButtonWasPushed(0);
   remoteControl- > off ButtonWasPushed(0);
   remoteControl- >onButtonWasPushed(1);
   remoteControl- > offButtonWasPushed(1);
   /*其余代码省略*/
   return 0;
}
```

【解析】

本题考查 Command 命令模式的概念及应用，并用 C++语言实现。Command 命令模式是一种对象行为模式，它主要解决的问题是，在软件构建过程中，“行为请求者”与“行为实现者”通常呈现一种“紧耦合”的问题。将一个请求封装为一个对象，从而可以用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作。

【答案】

(1) void execute(){}

(2) light->on()

(3) light->off()

(4) onCommands[slot]

(5) offCommands[slot]

(6) onCommands[slot]->execute()

(7) offCommands[slot]->execute()

【例 10-19】(2014 年 5 月下午试题五)

阅读下列说明和 C++代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

某实验室欲建立一个实验室环境监测系统，以显示实验室的温度、湿度以及洁净度等环境数据。当获取到最新的环境测量数据时，显示的环境数据能够更新。现在采用观察者(Observer)模式来开发该系统，观察者模式的类图如图 10-12 所示。

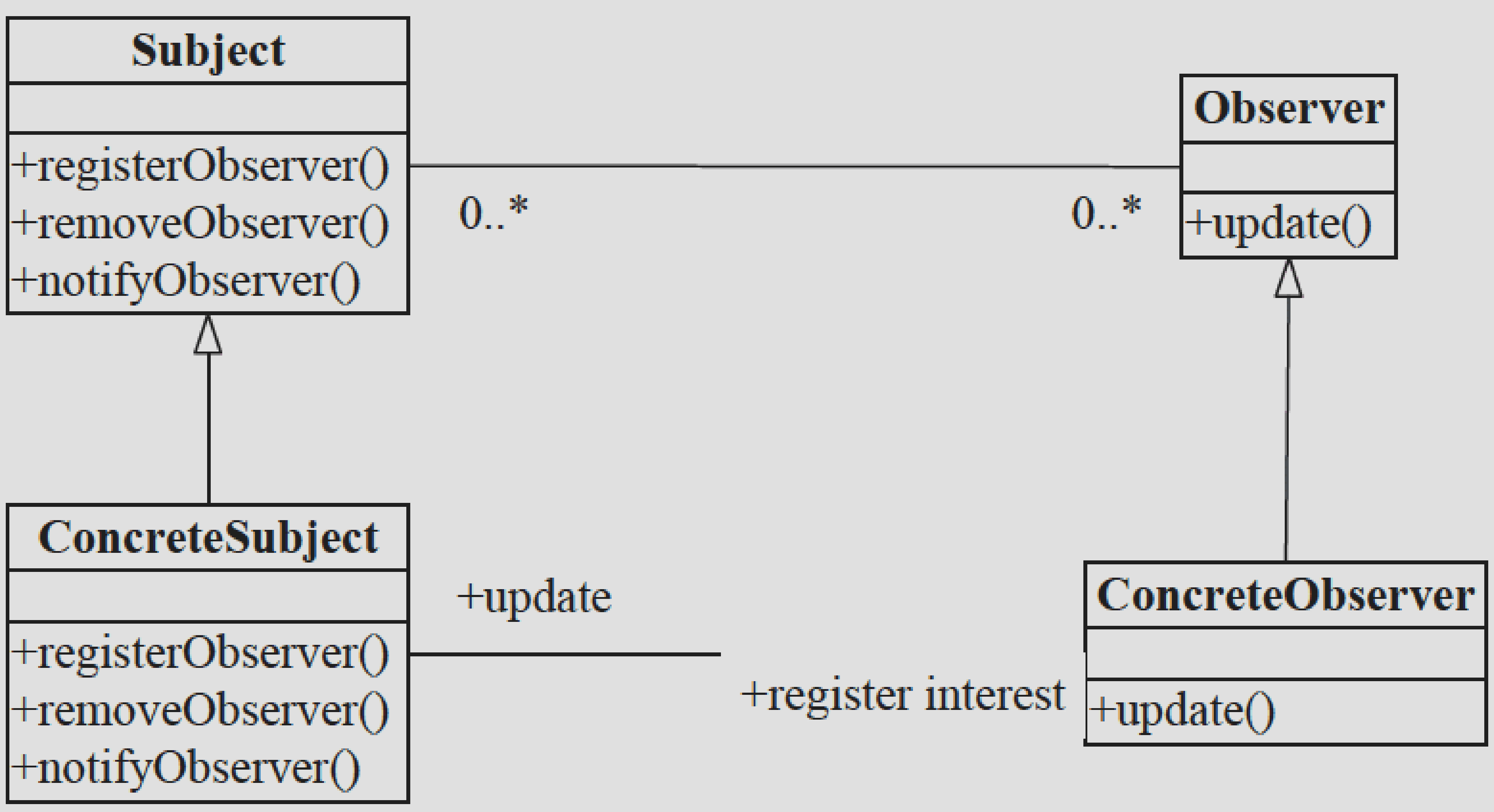


图 10-12　观察者模式的类图

【C++代码】

```
#include <iostream>
#include <vector>
using namespace std;
class Observer {
Public:
    virtual void update(float temp, float humidity, float cleanness)=0;
};
class Subject {
```

```
public:
    virtual void registerObserver(Observer*o)=0; //注册对主题感兴趣的观察者
    virtual void removeObserver(Observer*o)=0; //删除观察者
    virtual void notifyObservers()= 0;//当主题发生变化时通知观察者
}
class EnvironmentData : public___(1)___{
private:
    vector<Observer*> observers;
    float temperature, humidity, cleanness;
public:
    void registerObserver(Observer* o) { observers.push back(o); }
    void removeObserver(Observer*o)</*代码省略*/}
    void notifyObservers( ) {
        for(vector<Observe*>::const iterator it= observers.begin();it !=
        observers.end(); it++)
        {__(2)__; }
    }
    void measurementsChanged(){ __(3)__;}
    void setMeasurements(float temperature, float humidity, float cleanness){
        this->temperature= temperature;
        this->humidity = humidity;
        this->cleanness= cleanness;
        __(4)__;
    }
};
class CurrentConditionsDisplay : public__(5)__{
private:
    float temperature, humidity, cleanness;
    subject* envData;
public:
    CurrentConditionsDisplay(Subject* envData) {
        this->envData= envData;
         __(6)__:
    }
    void update(float temperature, float humidity, float cleanness)
    {   this->temperature=temperature;
        this->humidity = humidity;
        this->cleanness = cleanness;
        display( );
}
void display( ){/*代码省略*/}
};
int main( ){
    EnvironmentData* envData=new EnvironmentData0;
    CurrentConditionsDisplay* currentDisplay=new CurrentConditionsDisplay(envData);
    envData-> setMeasurements(80, 65, 30.4f);
    return 0;
}
```

【解析】当主题中的环境发生变化，也就是调用 measurementsChanged( )，会通知观察者，即调用

notifyObservers( )，在 notifyObservers()方法中，观查者会调用自身的 update(float temperature, float humidity, float cleanness)。在主题的 registerObserver(Observe*o)中会注册对主题感兴趣的观察者。

【答案】

(1) Subject

(2) (*it)->update(temperature,humidity,cleanness)

(3) notifyObservers()

(4) measurementsChanged()

(5) Observer()

(6) this->envData-> registerObserver(this)

【例 10-20】(2013 年 11 月下午试题五)

阅读以下说明和 C++代码，将应填入 (n) 处的字句写在对应栏内。

【说明】

欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。以绘制直线和圆形为例，对应的绘图程序如表 10-4 所示。

表 10-4　不同的绘图程序

	DP1	DP2
绘制直线	draw_a_line(x1,y1,x2,y2)	drawline(x1,x2,yl,y2)
绘制圆	draw_a_circle(x,y,r)	drawcircle(x,y,r)

该绘图软件的扩展性要求，将不断扩充新的图形和新的绘图程序。为了避免出现类爆炸的情况，现采用桥接(Bridge)模式来实现上述要求，得到如图 10-13 所示的类图。

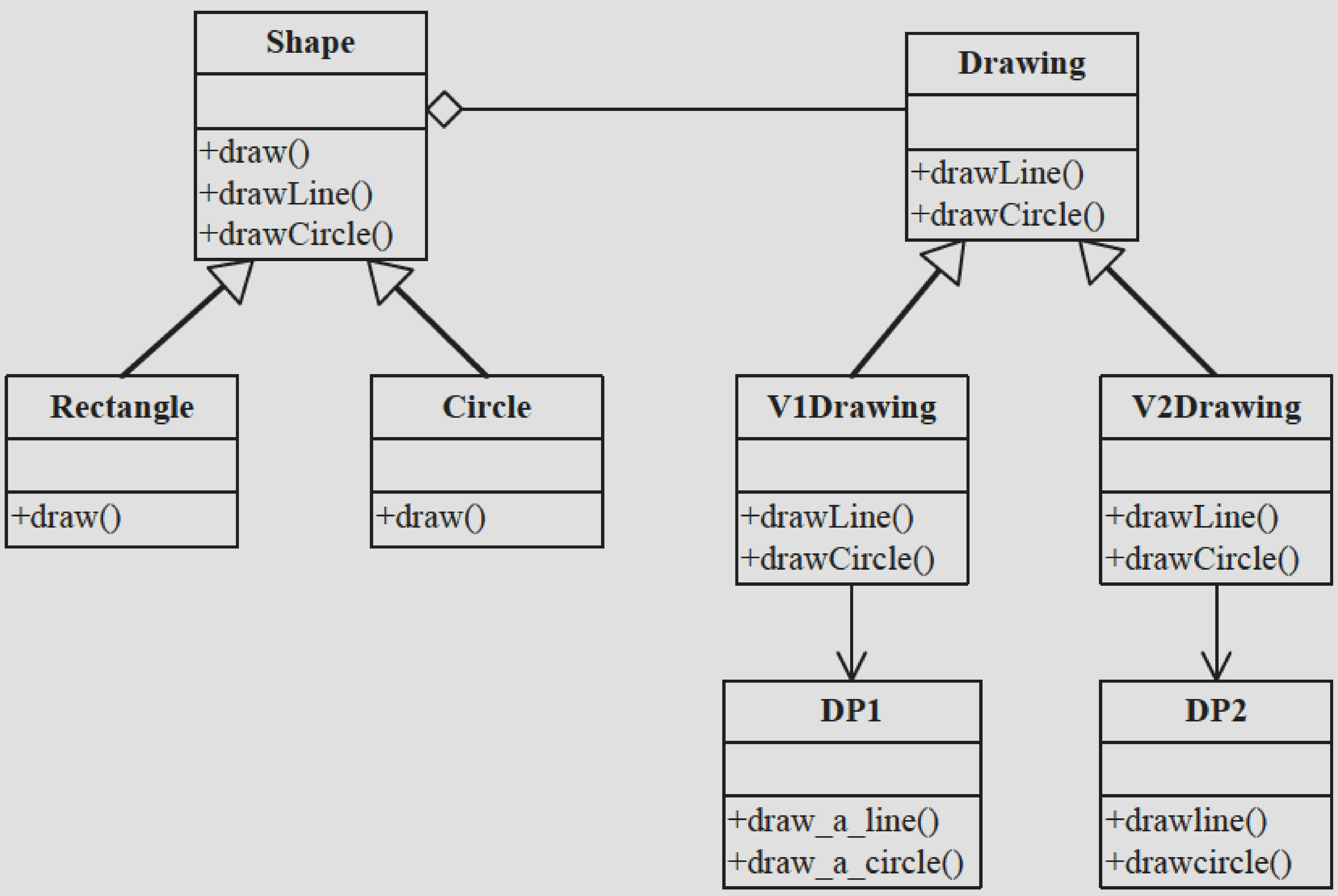


图 10-13　某绘图软件类图

【C++代码】

```
class DP1{
public:
    static void draw aline(double x1,double y1,double x2,double y2)
    {/* 代码省略 */}
    static void draw a circle(double x,double y,double r) {/* 代码省略 */}
};
class DP2(
public:
    static void drawline(double x1,double x2,double y1,double y2)
    {/* 代码省略 */}
    static void drawcircle(double x,double y,double r) {/* 代码省略 */}
};
class Drawing{
public:
    __(1)__;
    __(2)__;
};
class V1Drawing:public Drawing{
public:
    void drawLine(double x1,double y1,double x2,double y2) {/* 代码省略 */}
    void drawCircle(double x,double y,double r) {  __(3)__;  }
};
class V2Drawing:public Drawing{
public:
    void drawLine(double x1,double y1,double x2,double y2) {/* 代码省略 */}
    void drawCircle(double x,double y,double r) {  __(4)__;  }
};
class Shape{
public:
    __(5)__;
    Shape(Drawing *dp){ dp=dp;}
    void drawLine(double x1,double y1,double x2,double y2){
        dp->drawLine(x1,y1,x2,y2);}
    void drawCircle(double x,double y,double r){ dp->drawCircle(x,y,r);}
private:Drawing * dp;
};

class Rectangle:public Shape{
public:
    void draw(){   /* 代码省略 */   }
    //其余代码省略
};

class Circle:public Shape{
private: double  x, y, r;
public:
    Circle(Drawing *dp,double x,double y,double r): (6) {
 _x=x;_y=y;_r=r; }
```

```
        void draw(){    drawCircle( x, y, r);   }
    };
```

【解析】

本题考查 Bridge(桥接)模式的概念及应用，将抽象与其实现解耦，使它们都可以独立地变化。大致意思是说：将一组实现与另一组使用它们的对象分离。这里的实现指的是抽象类及其派生类用来实现自己的对象(而不是抽象类的派生类，这些派生类被称为具体类)。

Drawing 是一个虚拟基类，里面包含了希望不同策略实现的算法，派生类 V1Drawing、V2Drawing 都派生自 Drawing，对基类中的希望实现的算法都做了具体实现，且它们都含有 drawLine 和 drawCircle 函数，所以 Draw 类中缺失的应该是这两个算法，于是(1)处为 virtual void drawLine(double x1,double y1,double x2,double y2=0; (2)处为 virtual void drawCircle(double x,double y,double r)=0; DP1 和 DP2 中包含了绘制 Line 和 Circle 的具体实现的 Static 方法，因此在 V1Drawing、V2Drawing 类中可以直接调用它们，即(3)处为 DP1::draw_a_circle(x,y,r)；(4)处为 DP2.drawCircle(x,y,r); Shape 类派生出 Rectangle 和 Circle 类，里面都含有 draw 方法，但是具体 draw 方法的实现却不相同，所以(5)处为 virtual void draw()=0；最后 Circle 的构造函数初始化了后四个变量，还有继承自 shape 的变量_dp 未做初始化，因此调用 shape 的构造函数做初始化，即(6)处为 shape(dp)。

【答案】

(1) virtual void drawLine(double x1,double y1,double x2,double y2)=0;

(2) virtual void drawCircle(double x,double y,double r)=0;

(3) DP1::draw_a_ircle(x,y,r);

(4) DP2::drawCircle(x,y,r);

(5) virtual void draw()=0;

(6) Shape(dp)。

10.8　Java 程序设计

考核说明：本节主要介绍 Java 程序设计中的要点，要掌握类的继承、函数重载、抽象类的使用方法，这是下午必考内容。

10.8.1　基本概念

1. 应用领域

Java 目前主要应用于服务器端的企业级应用(Servlet、JSP)、手持设备(J2ME、K-Java、无线 Java)、普通网页(Applet)、普通应用程序。

2. 优点

Java 程序具有如下优点。

(1) 跨平台(大部分平台上都有 Java 虚拟机)，许多平台(计算机+操作系统)上都有各自的 Java 虚拟机(Java VM)，但 Java 虚拟机不跨平台，要分别编写。编译生成的是中间代码，

由统一的 Java 虚拟机指令组成。

(2) 代码可移动(与 HTML 相结合)。

(3) 完全面向对象。

(4) 编出来的程序不易出错(没有指针，内存垃圾自动回收，不会产生内存泄漏)。

此外，还有简单、安全、多线程等优点。

3. Java 与 C++的区别

Java 与 C++具有如下区别。

(1) 完全面向对象：无全局变量、无结构和联合、自动回收内存垃圾。

(2) 没有指针。

(3) 没有多继承。

(4) 解释执行。

10.8.2 基本语法

1. 注释

与 C 语言相同。多行用“/*…*/”，单行用//。

2. 基本数据类型

与 C 语言相同之处：char(8 bit)、short(16 bit)、int(32 bit)、long(64 bit)、float(32 bit)、double (64 bit)。

与 C 语言不同之处：byte(8 bit)，boolean(boolean 类型的变量取值为 true 或者 false)。

3. 常量

与 C 语言不同，Java 使用 final 关键字，C++语言则是 const，如 final float pi = 3.14f、final byte c = 12。浮点常数后面要加“f”，如 12.7f、1.02f。

4. 运算符

与 C 语言相同的有算术运算符、赋值运算符、逻辑运算符、比较运算符、自增自减运算符、位运算符和移位运算符。

5. 类型转换

与 C 语言相同，如 int s = (int)4.7f。

6. 基本语句

与 C 语言相同之处：if、switch、while、do…while、for。

输出函数 System.out.println 与 C 语言的 printf 函数不同，各输出项目之间用“+”连接，如 System.out.println("Id=" + nNum)。

10.8.3　程序设计

1. 类和继承

1)　类

一个类是一些属性和方法的封装体，类的定义用关键字 class 声明，用关键字 public、protected、private 指定类的成员的存取控制属性。private(私有)成员只有类内部的方法才能访问，protected(保护)成员可由派生类和同一文件夹下的类访问，public(公有)成员可以从类的外部访问，默认是 public。这体现了面向对象的以下指导思想：尽量将类内部的细节隐藏起来，对类的属性的操作应该通过类的方法来实现。

另外，public 还可以用来修饰类，public 类能够被其他文件夹下的类访问，非 public 类只能被同一文件夹下的类访问。一个.java 文件中，可以包含多个类，会被编译成多个.class 文件，但只能有一个 public 类，而且该类名要和文件名一样。

2)　继承

Java 中用关键字 extends 表示类间的继承关系。父类的公有属性和方法成为子类的属性和方法，子类如果有和父类同名、同参数类型的方法，那么子类对象在调用该方法时，调用的是子类的方法，亦即方法的重置。如果想要调用父类的同名方法，需要用 super 关键字(属性同理)。

子类的对象可以作为祖先类的对象使用，即所谓类的向上转换，反之则不行。具体表现在：可以用子类对象来对祖先类对象赋值，可以用子类对象作为实参去调用以父类对象为形参的函数。

2. 对象的引用本质

Java 中的对象实际上是对象的引用，本质上和 C 语言中的指针是一样的；但也和 C 语言指针不尽相同，比如，不能自增、自减，不能强制转换成其他类型。

3. 构造方法

构造方法就是类的对象生成时会被调用的方法。每个类至少有一个构造方法(Constructor)，也叫构造函数。构造方法的名称和类名相同，没有任何返回类型。每个类都有一个默认的构造方法，但当用户自定义了构造方法后，默认的构造方法就不再有效了。

4. 重载

同一个类中的两个或两个以上方法名称相同，而参数个数不同或参数类型不同，叫作重载。注意，如果各方法的名称和参数都一样，而仅仅返回值类型不同，则不是重载。

5. 静态属性和静态方法

静态属性和静态方法的声明用关键字 static 实现，一个类的静态属性只有一份，由所有该类的对象共享。不需要创建对象也能访问类的静态属性和方法，访问方式为“类名.静态属性或静态方法”，静态方法与对象无关，因此不能在静态方法中访问非静态属性和调用

非静态方法。

6. this 和 super 关键字

this 和 super 这两个关键字颇为重要。this 代表当前对象，super 代表当前对象的父类。

this 的主要用途有以下两个。

(1) 一个构造函数调用另一个构造函数，对构造函数的调用必须是第一条语句。

(2) 将对象自身作为参数来调用一个函数。

super 的用途：在子类中调用父类的同名方法，或在子类的构造函数中调用父类的构造函数，此时也必须是第一条语句。

7. 多态

所谓多态，是指通过基类对象调用一个基类和派生类都有的方法时，在运行时才能确定到底调用的是基类的方法还是派生类的方法。多态的好处是增加了程序的可扩展性。多态是通过动态联编实现的，即编译时不确定，程序运行时才确定调用哪个函数。

8. 抽象类与接口

1) 抽象类

抽象类通过关键字 abstract 实现，抽象类的目的是定义一个框架，规定某些类必须具有的一些共性。

包含抽象方法的类一定是抽象类，所谓抽象方法是指没有函数体的方法。

抽象类的直接派生类必须实现其抽象方法；抽象类只能用于继承，不能创建对象。

2) 接口

接口用关键字 interface 声明，只能用于继承。注意，此时关键字为 implements(实现)。

接口用于替代多继承的概念，能实现多继承的部分特点，又避免了多继承的混乱，还能起到规定程序框架的作用。注：接口也可用于多态。

直接继承了接口的类，必须实现接口中的抽象方法；间接的则可以实现，也可以不实现。

3) 抽象类与接口的异同

接口和抽象类都不能创建对象。

抽象类不能参与多继承，抽象类可以有非静态的成员变量，可以有非抽象方法；接口可以参与多继承，所有属性都是静态常量，所有方法都是 public 抽象方法。

9. final 关键字

用 final 关键字定义的常量，在其初始化或第一次赋值后，其值不能被改变。常量必须先有值，然后才能使用。对于常量的第一次赋值只能在构造函数中完成。

final 对象的值不能被改变，指的是该对象不能再指向其他对象，而不是指不能改变当前对象内部的属性值。

函数参数声明为 final 后，函数中不能改变其值。

final 方法是不能被重置的方法。

final 类不能被继承，其所有方法都是 final 的，但属性可以不是 final 的。

真题链接

【例 10-21】(2014 年 11 月下午试题六)

阅读下列说明和 Java 代码，将应填入(n)处的字句写在答题纸的对应栏内。

【说明】

某灯具厂商欲生产一个灯具遥控器，该遥控器具有 7 个可编程的插槽，每个插槽都有开关按钮，对应着一个不同的灯。利用该遥控器能够统一控制房间中该厂商所有品牌灯具的开关，现采用 Command(命令)模式实现该遥控器的软件部分。Command 模式的类图如图 10-14 所示。

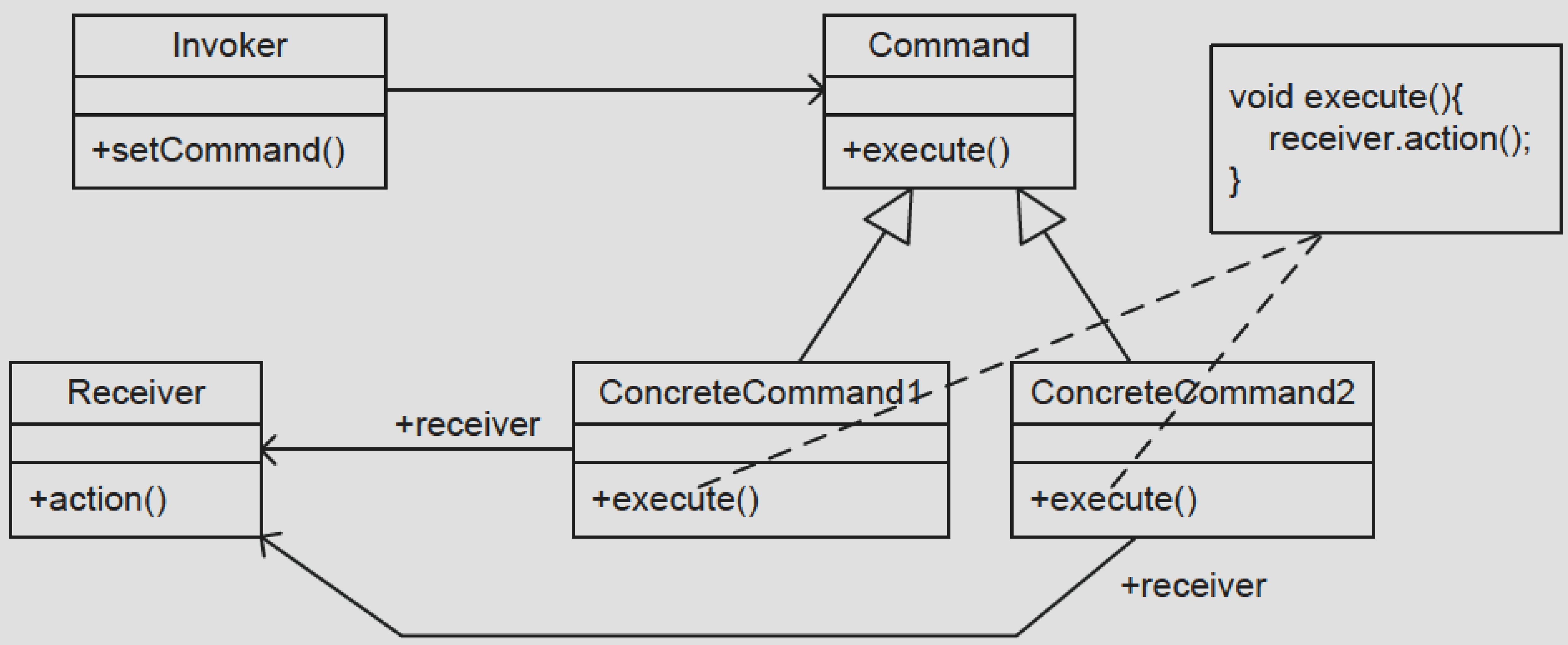


图 10-14　Command 模式类图

【Java 代码】

```
class Light{
   public Light(){};
   public Light(string name){/*代码省略*/}
   public void on(){/*代码省略*/}  //开灯
   public void off(){/*代码省略*/}  //关灯
其余代码省略
}
 (1) {
   public void execute() ;
}
class LightOnCommand implements Command{//开灯命令
   Light light;
   public LightOnCommand(Light light){this.light=light;}
   public void execute(){ (2) ;}
}
class LightOffCommand implements Command{//关灯命令
   Light light;
   public LightOffCommand(Light light){this.light=light;}
   public void execute(){ (3) ;}
}
class RemoteControl{ //遥控器
   Command[] onCommands[7];
   Command[] offCommands[7];
```

```
        public RemoteControl() { /*代码省略*/ }
        public void setCommand(int slot Command onCommand, Command offCommand) {
            (4) =onCommand;
            (5) =offCommand;
        }
        public void onButtonWasPushed(int slot) { (6);)
        public void offButtonWasPushed(int slot) { (7) ; }
    )
    class remoteLoader {
        public static void main (string[] args){
            RemoteControl remoteControl = new RemoteControl ();
            Light livingRoomLight=new Light("Living Room");
            Light kitchen Light= new Light("kitchen");
            LightOnCommand IivingRoomLightOn= new LightOnCommand(livingRoomLight);
            LightOffCommand livingRoomLightOff=new LightOffCommand(livingRoomLight);
            LightOnCommand kitchenLightOn=new LightOnCommand(kitchenLight);
            LightOffCommand kitchenLightOff= new LightOffCommand(kitchenLight);
            remoteControl.setCommand(0, livingRoomLightOn, livingRoomLightOff);
            remoteControl. setCommand(1, kitchenLightOn, kitchenLightOff);
            remoteControl.onButtonWasPushed(0);
            remoteControl. off ButtonWasPushed(0);
            remoteControl.onButtonWasPushed(1);
            remoteControl.offButtonWasPushed(1);
        }
    }
```

【解析】

本题考查 Command 命令模式的概念及应用，并用 Java 语言实现。Command 命令模式是一种对象行为模式。它主要解决的问题是，在软件的构建过程中，“行为请求者”与“行为实现者”通常呈现一种“紧耦合”的问题。将一个请求封装为一个对象，从而可用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤消的操作。

【答案】

(1) interface Command

(2) light.on()

(3) light.off()

(4) onCommands[slot]

(5) offCommands[slot]

(6) onCommands[slot].execute()

(7) offCommands[slot].execute()

【例 10-22】(2014 年 5 月上午试题六)

阅读下列说明和 Java 代码，将应填入(*n*)处的字句写在答题纸的对应栏内。

【说明】

某实验室欲建立一个实验室环境监测系统，能够显示实验室的温度、湿度以及洁净度等环境数据。当获取到最新的环境测量数据时，显示的环境数据能够更新。现在采用观察者(Observer)模式来开发该系统，

观察者模式的类图如图 10-15 所示。

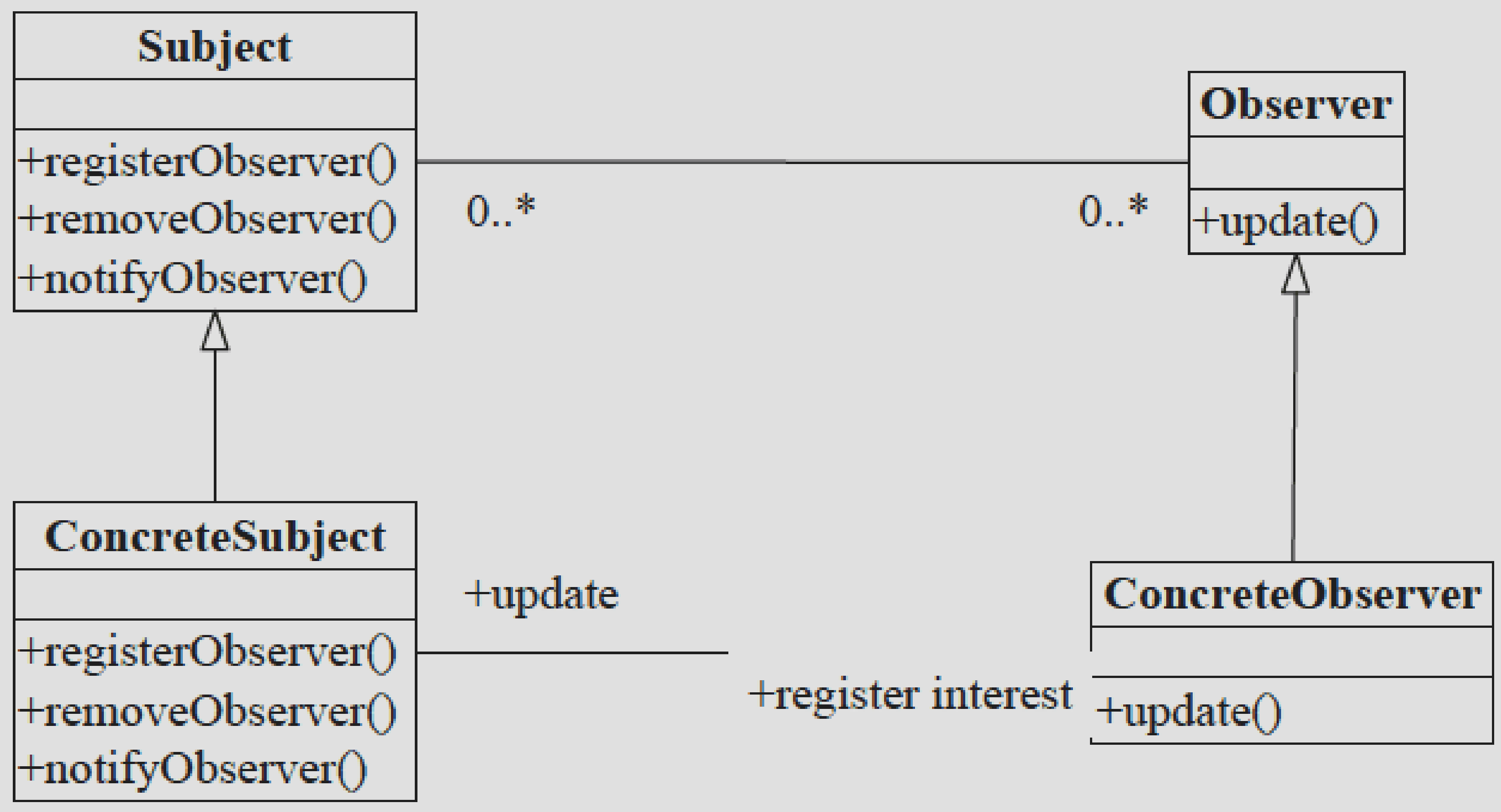


图 10-15　观察者模式的类图

【Java 代码】

```
import java.util.*
interface Observer {
    public void update(float temp, float humidity, float cleanness);
}
interface Subject {
    public void registerObserver(Observer o);//注册对主题感兴趣的观察者
    public void removeObserver(Observer o); //删除观察者
    public void notifyObservers()    //当主题发生变化时通知观察者
}
class EnvironmentData implements __(1)__ {
    private ArrayLjst observers;
    private float temperature, humidity, cleanness;
    public EnvironmentData()0 { observers = new ArrayList(); }
    public void registerObserver(Observer o) { observers.add(o); }
    public void removeObserver(Observer o)    {/*代码省略*/ }
    public void notifyObservers(){
        for (int i = 0; i < observers.size(); i++) {
            Observer observer=(Observer)observers.get(i);
            __(2)__ ;
        }
    }
    public void measurementsChanged(){ __(3)__ ;}
    public void setMeasurements(float temperature, float humidity, float
    cleanness) {
        this.temperature = temperature;
        this.humidity = humidity;
        this.cleanness = cleanness;
        __(4)__ ;
    }
}
class CurrentConditionsDisplay implements __(5)__{
    private float temperature;
    private float humidity;
```

```
    private float cleanness;
    private Subject envData;
    public CurrentConditionsDisplay(Subject envData) {
        this.envData=envData;
        __(6)__;
    }
    public void update(float temperature, float humidity, float cleanness) {
        this.temperature=temperature;
        this.humidity = humidity;
        this.cleanness = cleanness;
        display();
    }
    public void display(){/*代码省略*/}
}
class EnvironmentMonitor{
    public static void main(String[] args) {
        EnvironmentData envData=new EnvironmentData();
        CurrentConditionsDisplay currentDisplay= new
        currentConditionsDisplay(envData);
        envData.setMeasurements(80, 65 , 30.4f);
    }
    public void display(){/*代码省略*/}
}
class EnvironmentMonitor{
    public static void main (String[] args) {
        EnvironmentData envData= new EnvironmentData();
        CurrentConditionsDisplay currentDisplay= new
        currentConditionsDisplay(envData);
        envData.setMeasurements(80, 65 , 30.4f);
    }
}
```

【解析】EnvironmentData 是环境数据，也就是我们要监测的对象，即主题(Subject)，因此(1)处为 Subject，(2)处为通知观察者，因此遍历观察者容器，遍历到一个观察者对象，则更新该观察者的数据，即调用观察者的 update()方法，当环境数据变化时，需要通知观察者，因此(4)处是调用环境变化方法 measurementsChanged()，通过此方法通知观察者更新数据，因此，(3)处为 notifyObservers()。

根据 CurrentConditionsDisplay 类中的 update()方法可知：CurrentConditionsDisplay 是个观察者，因此(5)处为 Observer，(6)是将观察者添加到主题中去。

【答案】

(1) Subject

(2) observer.update(temperature,humidity,cleanness)

(3) notifyObservers()

(4) measurementsChanged()

(5) Observer()

(6) envData.registerObserver(this)

【例 10-23】(2013 年 11 月下午试题六)

阅读以下说明和 Java 代码，将应填入__(n)__处的字句写在对应栏内。

【说明】

欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。以绘制直线和圆形为例，对应的绘图程序如表 10-5 所示。

表 10-5　不同的绘图程序

	DP1	DP2
绘制直线	draw_a_line(x1,y1,x2,y2)	drawline(x1,x2,yl,y2)
绘制圆	draw_a_circle(x,y,r)	drawcircle(x,y,r)

该绘图软件的扩展性要求，将不断扩充新的图形和新的绘图程序。为了避免出现类爆炸的情况，现采用桥接(Bridge)模式来实现上述要求，得到如图 10-16 所示的类图。

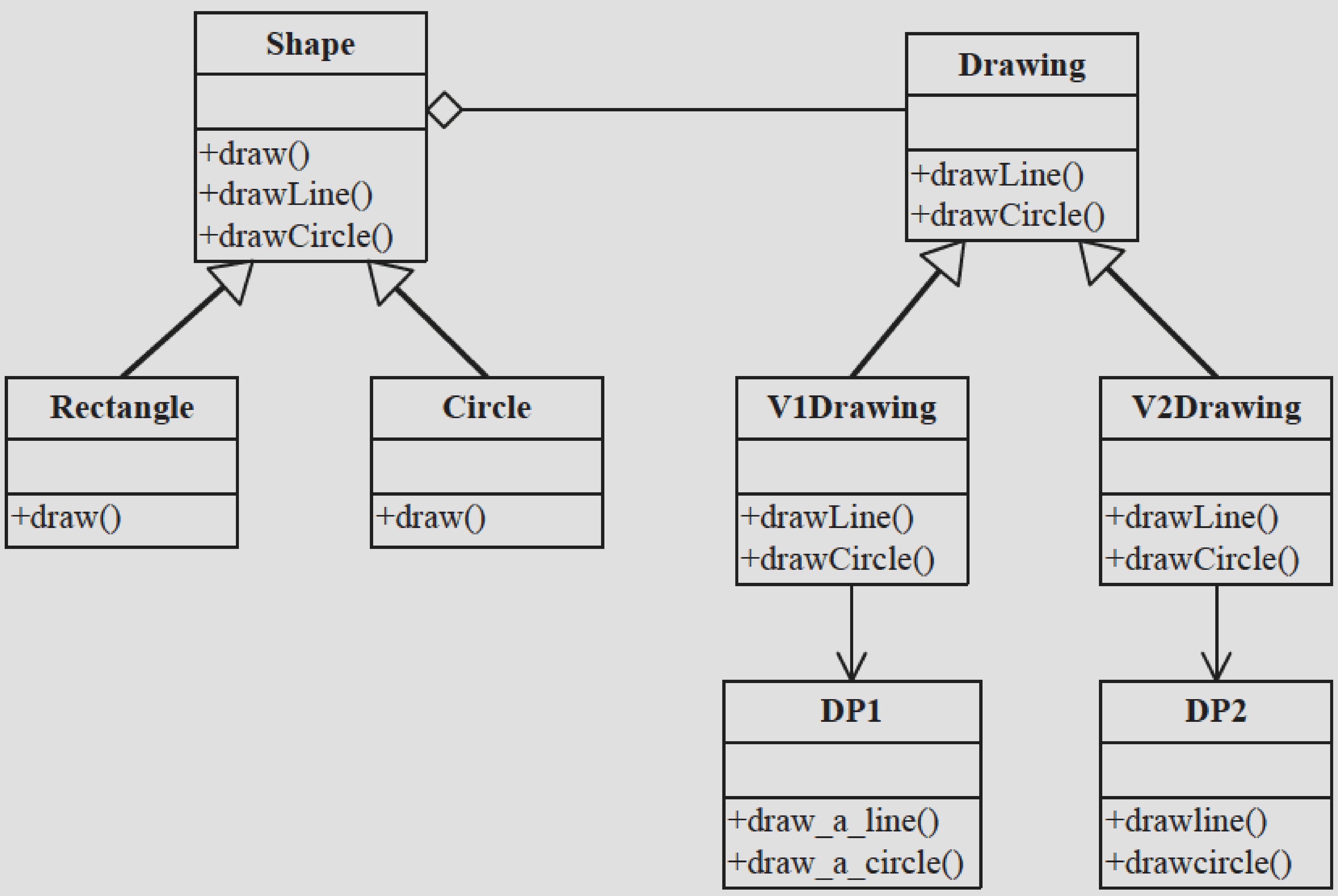


图 10-16　某绘图软件类图

【Java 代码】

```
__(1)__  Drawing{
    __(2)__ ;
      (3)   ;
}

class DP1{
    static public void draw aline(double x1,double y1,double x2,double y2)
        {/* 代码省略 */}
    static public void draw a circle(double x,double y,double r)
        {/* 代码省略 */}
};

class DP2(
    static public void drawline(double x1,double x2,double y1,double y2)
```

```
        {/* 代码省略 */}
    static public void drawcircle(double x,double y,double r) {/* 代码省略 */}
};

class V1Drawing implements Drawing{
    public void drawLine(double x1,double y1,double x2,double y2)
        {/* 代码省略 */}
    public void drawCircle(double x,double y,double r) {     (4)  ;    }
};

class V2Drawing implements Drawing{
    public void drawLine(double x1,double y1,double x2,double y2)
        {/* 代码省略 */}
    public void drawCircle(double x,double y,double r) {    __(5)__;    }
};

abstract class Shape{
    private Drawing  dp;
    __(6)__;
    Shape(Drawing dp){ dp=dp;}
    public void drawLine(double x1,double y1,double x2,double y2){
         dp.drawLine(x1,y1,x2,y2); }
    public void drawCircle(double x,double y,double
    r){ dp.drawCircle(x,y,r);}
};

class Rectangle extends Shape{
    private double  x1, x2, y1, y2;
    public Rectangle(Drawing dp,double x1,double y1,double x2,double y2)
        {/* 代码省略 */}
    public void draw(){ /* 代码省略 */    }
    };

class Circle extends Shape{
    private double  x, y, r;
    public Circle(Drawing dp,double x,double y,double r) { /* 代码省略 */
}
    public void draw(){ drawCircle( x, y, r);   }
};
```

【解析】

本年的试题 6 和试题 5 其实是相同的，只是实现方法不同，本题考查 Bridge(桥接)模式的概念及应用，将抽象与其实现解耦，使它们都可以独立地变化。大致意思是说：将一组实现与另一组使用它们的对象分离。Java 语言中声明抽象函数用 abstract，且每个函数都需要声明访问类型。对照试题 5 的代码我们可以看出：(1)处为 interface。

这里的实现指的是抽象类及其派生类用来实现自己的对象(而不是抽象类的派生类，这些派生类被称为具体类)Drawing 是一个虚拟基类，里面包含了希望不同策略实现的算法，派生类 V1Drawing、V2Drawing 都派生自 Drawing，对基类中的希望实现的算法都做了具体实现，且它们都含有 drawLine 和 drawCircle 函数，所以 Draw 类中缺失的应该是这两个算法，于是(2)处为 void drawLine(double x1,double y1,double x2,double y2)，(3)处为 void drawCircle(double x,double y,double r)；DP1 和 DP2 中包含了绘制 Line 和 Circle 的具体实现的 Static 方法，因此在 V1Drawing、V2Drawing 类中可以直接进行调用它们，因此(4)处为

DP1.draw_a_circle(x,y,r)，(5)处为 DP2.drawCircle(x,y,r)；Shape 类派生出 Rectangle 和 Circle 类，里面都含有 draw 方法，但是具体 draw 方法的实现却不相同，所以(6)处为 abstract public void draw。

【答案】

(1) interface

(2) void drawLine(double x1,double y1,double x2,double y2)

(3) void drawCircle(double x,double y,double r)

(4) DP1.draw_a_circle(x,y,r);(5) DP2.drawCircle(x,y,r)

(6) abstract public void draw()

10.9　应试加油站

10.9.1　考频统计

通过对近几年考试真题的分析，“面向对象技术”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 10-6 所示。

表 10-6　历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：37～47	面向对象程序设计语言、类、多态、对象、UML 事物、UML 交互图、设计模式	11
	下午：试题三、试题五或试题六	用例、对应的类名、对应的多重度、类图的修改 Command 模式的概念及 C++/Java 程序设计	30
2014 年 5 月	上午：37～47	面向对象程序设计语言、类、绑定、多态、UML 状态图、UML 序列图、设计模式	11
	下午：试题三、试题五或试题六	用例、对应的类名、方法名、设计模式分析 观察者模式的概念及 C++/Java 程序设计	30
2013 年 11 月	上午：37～47	领域类模型、UML 活动图、UML 序列图、面向对象的基本概念、策略设计模式、状态设计模式、组件设计模式	11
	下午：试题三、试题五或试题六	完善状态图和类图、设计模式分析 桥接模式的概念及 C++/Java 程序设计	30
2013 年 5 月	上午：37～47	多态性、继承、UML 部署图、UML 状态图、UML 中类图、设计模式	11
	下午：试题三、试题五或试题六	完善用例图和类图、选择、删除候选类的原则 原型模式的概念及 C++/Java 程序设计	30
2012 年 11 月	上午：37～47	对象、继承、多态、部署图、概念图、UML 接口、活动图、状态模式、单例模式	11
	下午：试题三、试题五或试题六	用例图和类图的名称确定、类图的修改 抽象工厂模式和 C++/Java 程序设计	**30**
2012 年 5 月	上午：37～47	组合关系、封装性、动态绑定、UML 序列图、适配器模式、组合	11
	下午：试题三、试题六或试题七	用例图和类图的名称确定、多重度、类图的属性 装饰器(Decorator)模式和 C++/Java 程序设计	30

纵观历年试卷，本章知识点是以选择题和综合分析题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量大约为 11 道选择题，所占分值为 11 分(约占试卷总分值 75 分中的 15%)；在下午试卷中，所考查的题量为 2 道综合分析题，所占分值为 30 分(约占试卷总分值 75 分中的 40%)。本章试题理论与实践应用并重，难度中等偏难。C++程序设计题、Java 程序设计题在考试中二选一，要重点掌握其中一门面向对象程序设计语言。UML 面向对象分析是下午科目必考题目，要重点掌握。

10.9.2 解题技巧

【典型题 10-1】在多态的几种不同形式中，__(37)__多态是一种特定的多态，指同一个名字在不同上下文中可代表不同的含义。(2013 年 5 月真题 37)

(37) A. 参数　　B. 包含　　C. 过载　　D. 强制

【解析】一般将多态分为通用多态和特殊多态。其中通用多态包括参数多态和包含多态，参数多态利用泛型编程，是发散式的，是静态绑定的，让相同的实现代码应用于不同的场合，看重的是算法的普适性；包含多态利用 OOP，是收敛式的，是动态绑定的，让不同的实现代码应用于相同的场合，看重的是接口与实现的分离度。特殊多态包括强制多态和过载多态，其中强制多态即一种类型的变量在作为参数传递时隐式转换成另一种类型，比如一个整型变量可以匹配浮点型变量的函数参数；过载多态即同一个名(操作符、函数名)在不同的上下文中有不同的类型。

【答案】C

【典型题 10-2】继承是父类和子类之间共享数据和方法的机制。以下关于继承的叙述中，不正确的是__(38)__。有关图 10-17 中 doIt()方法的叙述中，正确的是__(39)__。(2013 年 5 月真题 38～39)

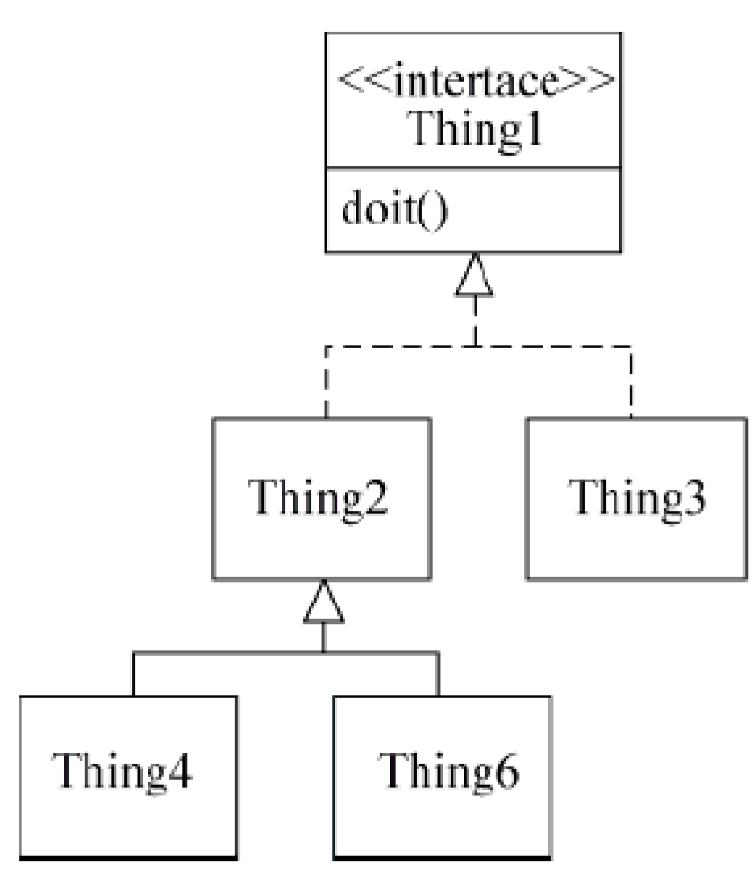


图 10-17　类图

(38) A. 一个父类可以有多个子类，这些子类都是父类的特例
　　B. 父类描述了这些子类的公共属性和操作
　　C. 子类可以继承它的父类(或祖先类)中的属性和操作而不必自己定义
　　D. 子类中可以定义自己的新操作而不能定义和父类同名的操作

(39) A. doIt()必须由 Thing3 实现，同时可能由 Thing4 实现
　　B. doIt()必须由 Thing5 实现
　　C. doIt()必须由 Thing2、Thing3、Thing4 和 Thing5 实现

D. doIt()已经由 Thing1 实现，因此无须其他类实现

【解析】继承是父类和子类之间共享数据和方法的机制。这是类之间的一种关系，在定义和实现一个类(子类)的时候，可以在一个已经存在的类(父类)的基础上进行，把这个已经存在的类所定义的内容作为自己的内容，并加入若干新的内容。一个父类可以有多个子类，这些子类都是父类的特例，父类描述了这些子类的共有属性和操作。一个子类可以继承它的父类(或祖先类)中的属性和操作，这些属性和操作在子类中不必定义，子类中还可以定义自己的属性和操作。所以 D 选项错误。

题中的 Thing1 为接口，那么 doIt()为接口中的抽象方法，必须由实现它的类去实现该方法。因此在 Thing3 中必须实现，而 Thing4 也是 Thing1 的子类，但不是直接子类，所以可能由 Thing4 实现。

【答案】(38)D；(39)A

【典型题 10-3】在面向对象技术中，__(41)__定义了超类和子类之间的关系，子类中以更具体的方式实现从父类继承来的方法称为__(42)__，不同类的对象通过__(43)__相互通信。(2013 年 5 月真题 41～43)

(41) A. 覆盖　　B. 继承　　C. 消息　　D. 多态

(42) A. 覆盖　　B. 继承　　C. 消息　　D. 多态

(43) A. 覆盖　　B. 继承　　C. 消息　　D. 多态

【解析】本题主要考察面向对象的基本概念。继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。

【答案】(41)B；(42)A；(43)C

【典型题 10-4】在面向对象技术中，对象具有以下特性：__(37)__。(2012 年 11 月真题 37)

①清晰的边界 ②良好定义的行为 ③确定的位置和数量 ④可扩展性

(37) A. ②④　　B. ①②③④　　C. ①②④　　D. ①②

【解析】 对象是类的实例，具有清晰的边界、良好定义的行为和可扩展性。

【答案】C

【典型题 10-5】在面向对象技术中，__(38)__说明一个对象具有多种形态，__(39)__定义超类与子类之间的关系。(2012 年 11 月真题 38～39)

(38) A. 继承　　B. 组合　　C. 封装　　D. 多态

(39) A. 继承　　B. 组合　　C. 封装　　D. 多态

【解析】在面向对象技术中，不同的对象收到同一消息可以产生完全不同的结果，这一现象叫作多态，这说明一个对象具有多种形态；继承是父类和子类之间共享数据和方法的机制，这是类之间的一种关系。

【答案】(38)D；(39)A

【典型题 10-6】以下关于封装在软件复用中所充当的角色的叙述，正确的是__(38)__。(2012 年 5 月真题 38)

(38) A. 封装使得其他开发人员不需要知道一个软件组织内部是如何工作的

B. 封装使得软件组织更有效地工作

C. 封装使得软件开发人员不需要编制开发文档

D. 封装使得软件组件开发更加容易

【解析】封装就是将抽象得到的数据和行为(或功能)相结合，形成一个有机的整体，也就是将数据与操作数据的源代码进行有机结合，形成“类”，其中数据和函数都是类的成员。封装的目的是增强安全性和简化编程，使用者不必了解具体的实现细节，而只是要通过外部接口，以特定的访问权限来使用类的

成员。

【答案】A

【典型题 10-7】在有些程序设计语言中，过程调用和响应调用需执行的代码的绑定直到运行时才进行，这种绑定称为 (39) 。(2012 年 5 月真题 39)

(39) A. 静态绑定　　B. 动态绑定　　C. 过载绑定　　D. 强制绑定

【解析】绑定指的是一个方法的调用与方法所在的类(方法主体)关联起来。

静态绑定：在程序执行前方法已经被绑定，此时由编译器或其他连接程序实现。

动态绑定：在运行时根据具体对象的类型进行绑定。若一种语言实现了动态绑定，同时必须提供一些机制，可在运行期间判断对象的类型，并分别调用适当的方法。也就是说，编译器此时依然不知道对象的类型，但方法调用机制能自己去调查，找到正确的方法主体。不同的语言对动态绑定的实现方法是有所区别的。

【答案】B

【典型题 10-8】以下关于 UML 部署图的叙述中，正确的是 (40) 。(2013 年 5 月真题 40)

(40) A. 因为一条消息总是有某种响应，部署组件之间的依赖是双向的

B. 部署组件之间的依赖关系类似于包图

C. 部署图不用于描述代码的物理模块

D. 部署图不用于描述系统在不同计算机系统的物理分布

【解析】部署图展现了运行处理节点以及其中的构件的配置。部署图给出了体系结构的静态实施视图。它与构件图相关，通常一个节点包含一个或多个构件即类似于包图，所以答案选 B。

【答案】B

【典型题 10-9】以下关于 UML 状态图的叙述中，不正确的是 (41) 。对图 10-18 的描述正确的是 (42) 。(2013 年 5 月真题 41～42)

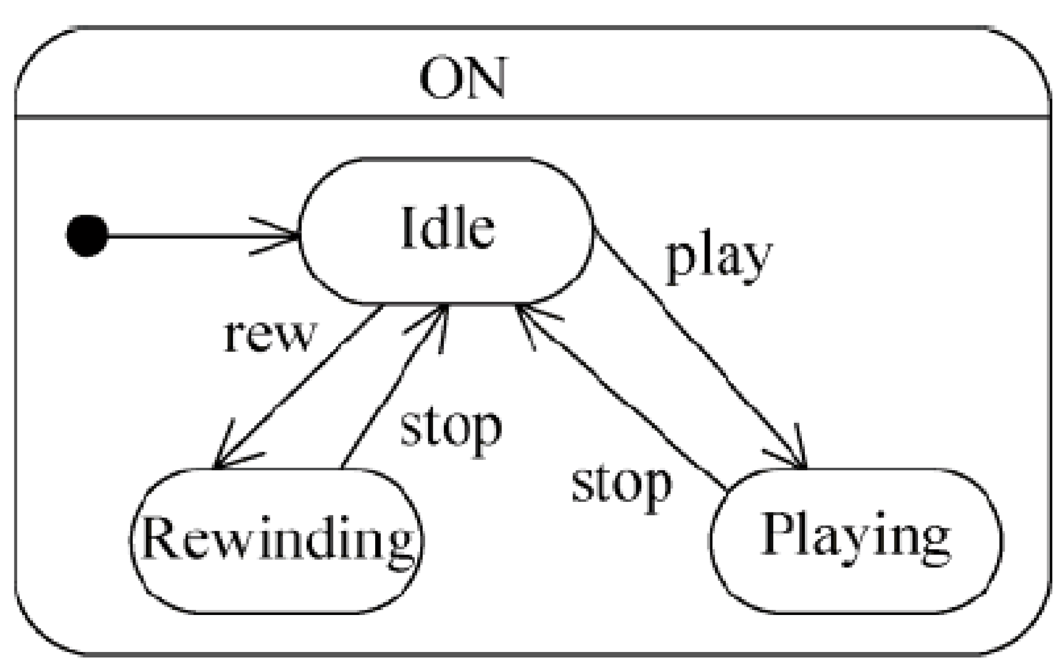


图 10-18　UML 状态图

(41) A. 用于描述一个对象在多个用例中的行为

B. 用于某些具有多个状态的对象而不是系统中大多数或全部对象

C. 用于描述多个对象之间的交互

D. 可用于用户界面或控制对象

(42) A. ON 是一个并发状态

B. 因为此状态图中没有终止(final)状态，所以此图是无效的

C. play、stop 和 rew 是动作

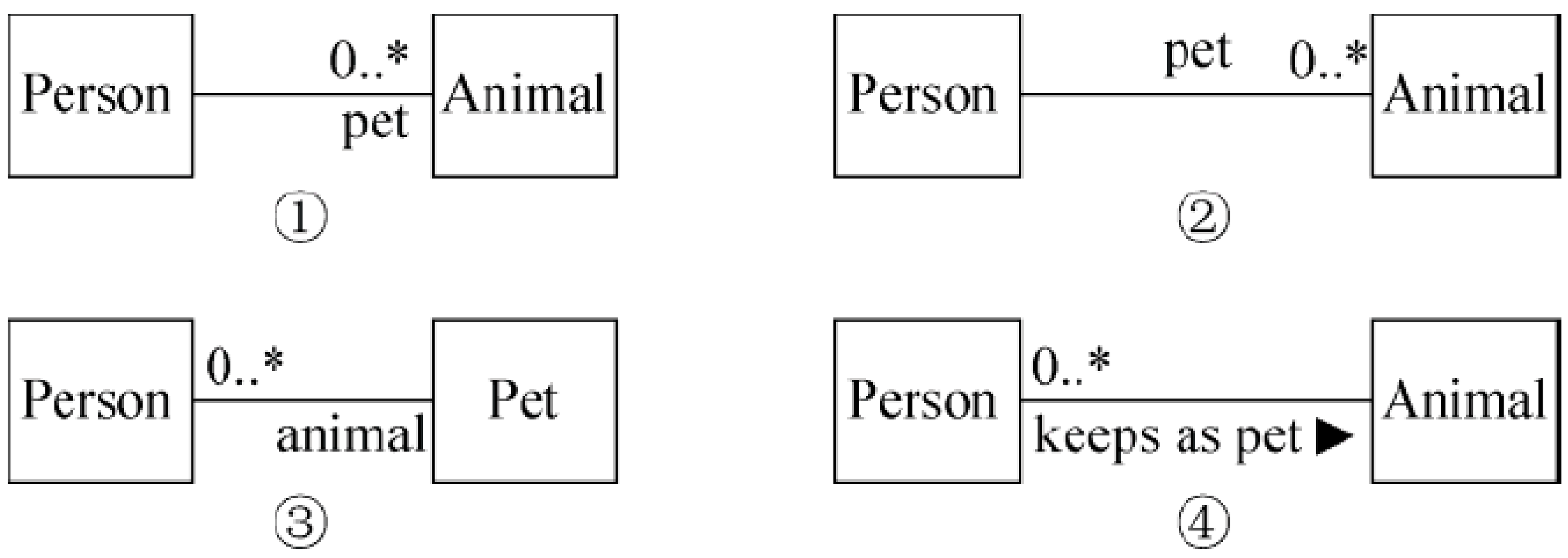
D. ON 是超状态

【解析】状态图展现了一个状态机，它由状态、转换、事件和活动组成。状态图关注系统的动态视图，

它对接口、类和协作的行为建模尤为重要，它强调对象行为的事件顺序。状态图通常包含简单状态和组合状态、转换(事件和动作)。可以用状态图对系统的动态方面建模。这些动态方面可以包括出现在系统体系结构的任何视图中的任何一种对象的按事件排序的行为，这些对象包括类(主动类)、接口、构件和节点。所以状态图不表示多个对象之间的交互。根据 ON 状态的内部行为可以发现该状态为超状态。

【答案】(41)C；(42)D

【典型题 10-10】描述一些人(Person)将动物(Animal)养为宠物(Pet)的是图 __(43)__ 。(2013 年 5 月真题 43)



(43) A. ① B. ② C. ③ D.④

【解析】该题考查 UML 中类图的表示，由 Person 类和 Animal 类之间一比多的关系以及类图的画法可知选项为 A。

【答案】A

【典型题 10-11】在领域类模型中不包含 __(37)__ 。(2013 年 11 月真题 37)

(37) A. 属性 B. 操作 C. 关联 D. 领域对象

【解析】领域模型是对领域内的概念类或现实世界中对象的可视化表示。领域类模型从按对象分类的角度来创建对象领域的描述，包括定义概念、属性和重要的关联，其结果用一组显示领域概念和对象的图形——类图来组织。

【答案】D

【典型题 10-12】在执行如图 10-19 所示的 UML 活动图时，能同时运行的最大线程数为 __(38)__ 。(2013 年 11 月真题 38)

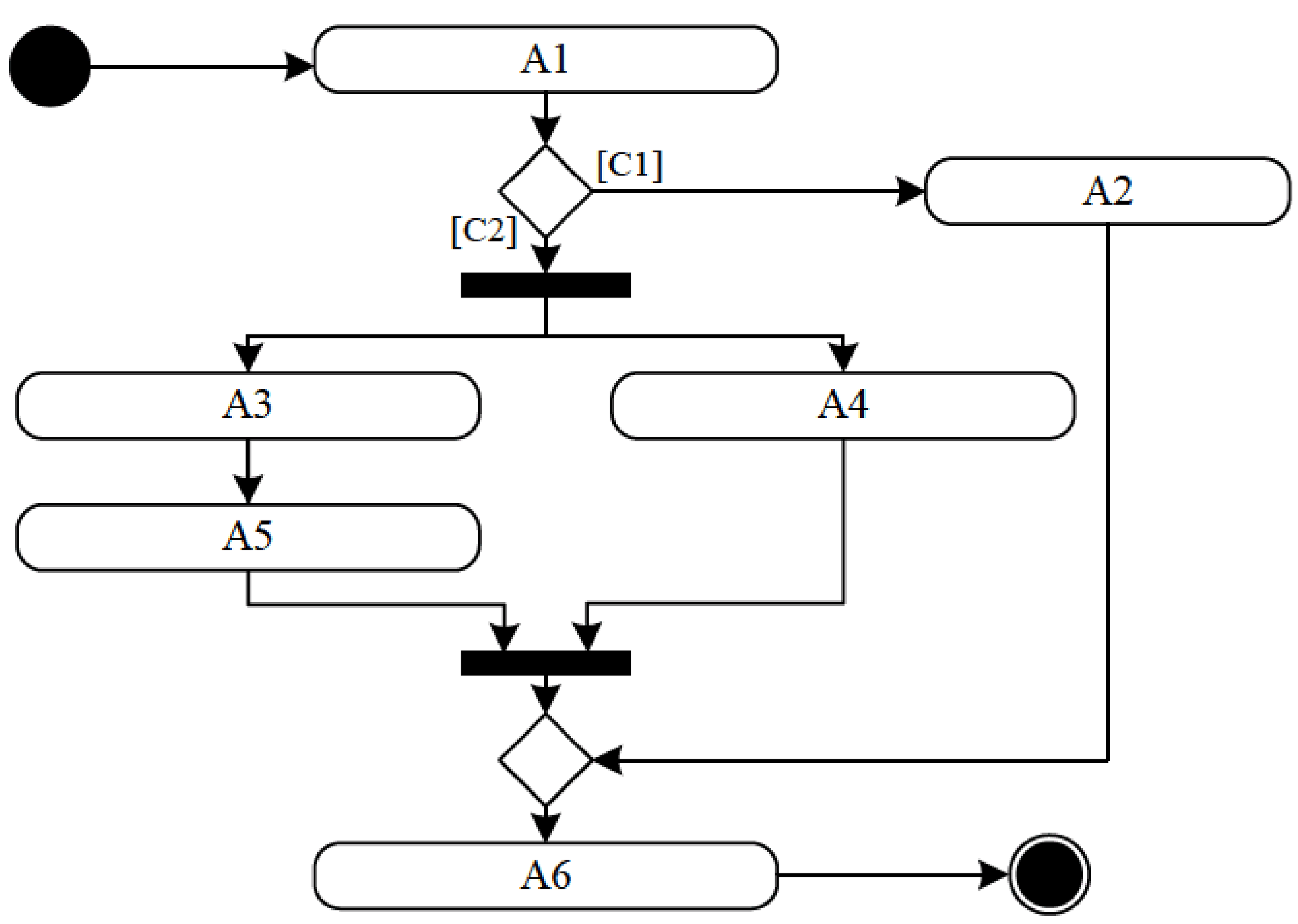


图 10-19 某 UML 活动图

(38) A. 4　　B. 3　　C. 2　　D. 1

【解析】从 UML 活动图中可以看出能够同时运行的最大线程为 A3、A4，答案选 C。

【答案】C

【典型题 10-13】如图 10-20 所示的 UML 序列图中，__(39)__表示返回消息，Account 应该实现的方法有__(40)__。(2013 年 11 月真题 39～40)

(39) A. xfer　　B. check　　C. evaluation　　D. minus

(40) A. xfer( )　　B. xfer( )、plus( )和 minus( )

C. check( )、plus( )和 minus( )　　D. xfer( )、evaluation( )、plus( )和 minus( )

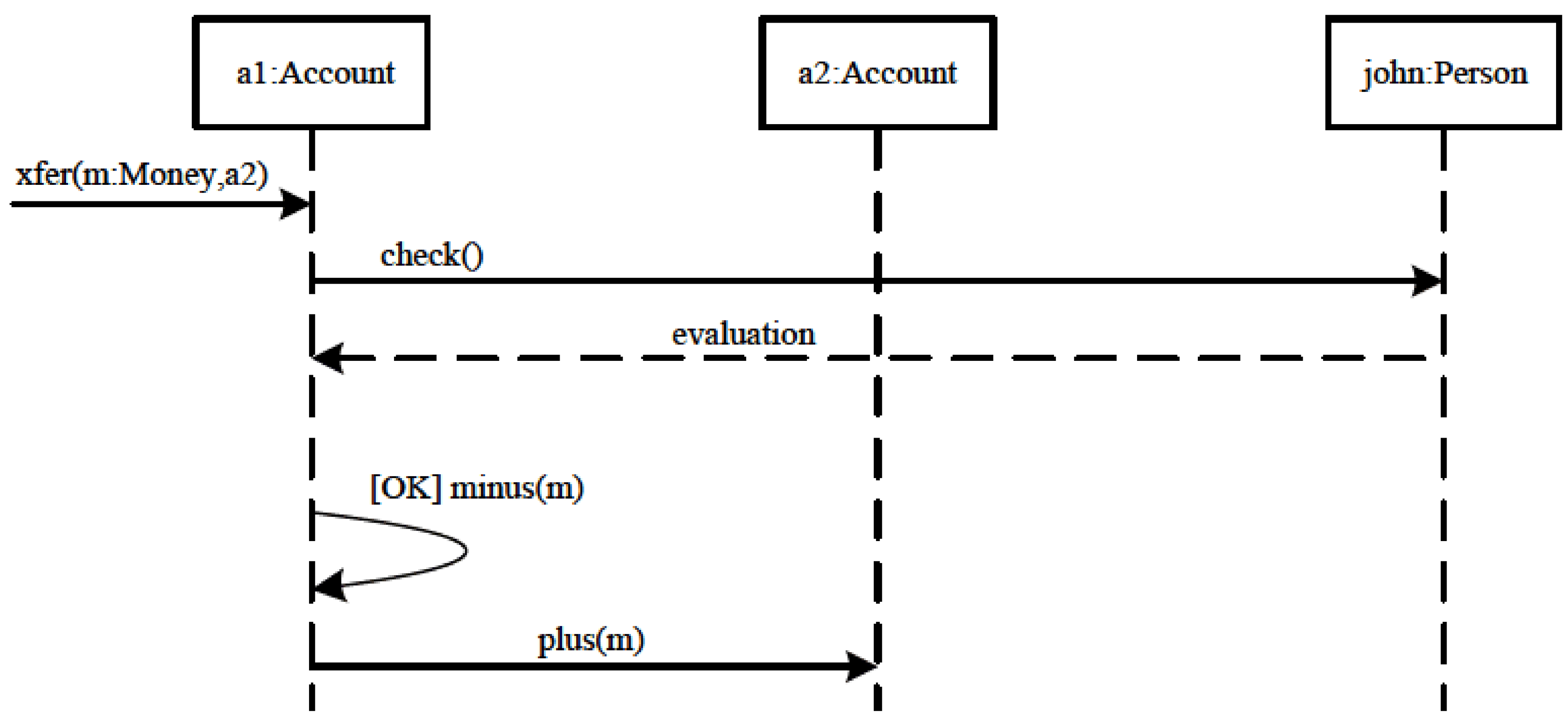


图 10-20　某 UML 序列图

【解析】一个返回消息画作一个带开放箭头的虚线，向后指向来源的生命线，在这条虚线上面，放置操作的返回值，所以说返回消息为 evaluation。从图中可见，a1、a2 均为 Account 对象，需要实现 xfer( )、minus( )和 plus( )方法。Person 应该实现 check( )方法。

【答案】(39)C；(40)B

【典型题 10-14】如果要表示待开发软件系统中软件组件与硬件之间的物理关系，通常采用 UML 中的__(40)__。(2012 年 11 月真题 40)

(40)A. 组件图　　B. 部署图　　C. 类图　　D. 网络图

【解析】UML 提供了多种视图，其中部署图展现了运行处理节点以及其中构件的配置。部署图给出了体系结构的静态实施视图，表示待开发软件系统中软件组件与硬件之间的物理关系，通常在系统实施阶段使用。

【答案】B

【典型题 10-15】对于场景：一个公司负责多个项目，每个项目(Project)由一个员工(Employee)团队(Team)来开发。下列 UML 概念图(见图 10-21)中，__(41)__最适合描述这一场景。(2012 年 11 月真题 41)

(41) A. 图 A　　B. 图 B　　C. 图 C　　D. 图 D

【解析】 在 UML 图中，关联表示对象之间的连接，在关联上可以标注重复度和角色。选项 A 中，项目(Project)是直接与员工(Employee)建立关系的错误。选项 C 中，Company 和 Project 之间的关系和重复度不正确；选项 D 中，Company 和 Project 之间不存在泛化关系；只有 B 选项最适合描述题目所述的场景。

【答案】B

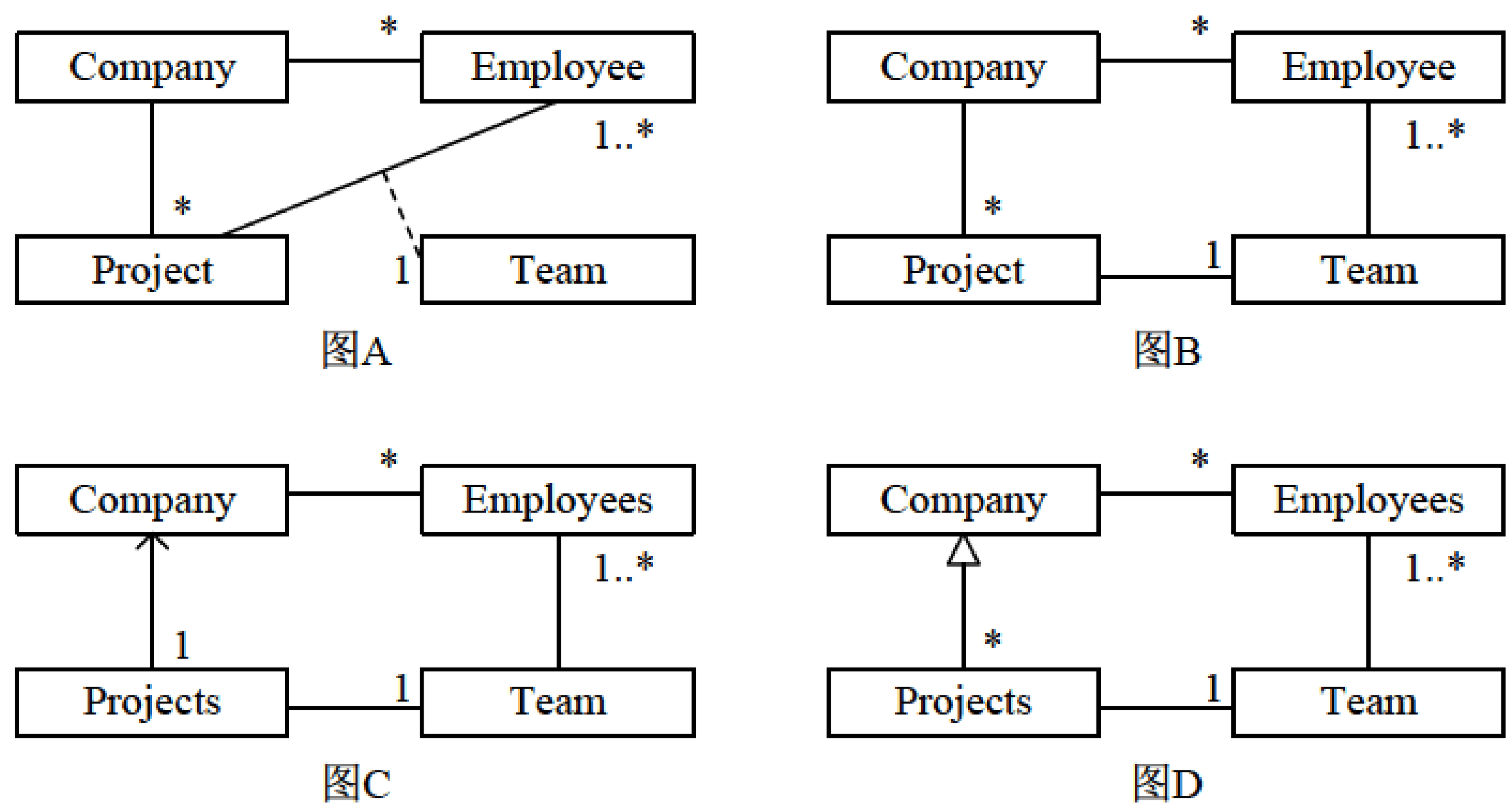


图 10-21　UML 概念图

【典型题 10-16】UML 中接口可用于(42)。(2012 年 11 月真题 42)

(42) A. 提供构造型(stereotype)名称为<<interface>>的具体类

B. Java 和 C++程序设计中，而 C#程序设计中不支持

C. 定义可以在多个类中重用的可执行逻辑

D. 声明对象类所需要的服务

【解析】接口(interface)是描述类的部分行为的一组操作，也是一个类提供给另一个类的一组操作。和类的不同之处在于，接口只是一组操作，没有属性。要将类和接口区分开来，一种办法是使用构造型<<interface>>，把它放在矩形框中接口的名字之上，另一种办法是接口的名字以大写字母“I”开头。接口可用于 Java 和 C#程序设计中，而 C++程序设计中不支持。

【答案】D

【典型题 10-17】图 10-22 活动图中可以同时执行的活动是(43)。(2012 年 11 月真题 43)

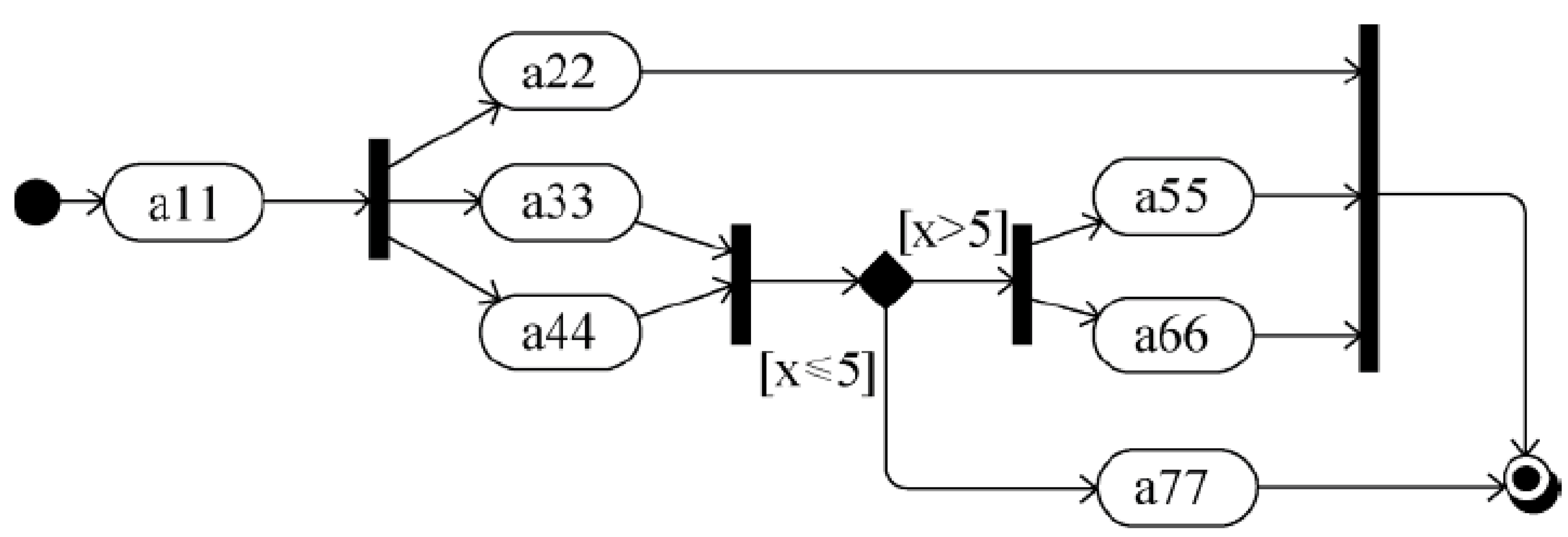


图 10-22　活动图

(43) A. a44 和 a66　　B. a22、a33 和 a44　　C. a11 和 a77　　D. a66 和 a77

【解析】活动图的主要要素包括初始节点、活动节点、活动终点、转换、分支与监护条件、分岔与汇合等。在该活动图中，a11～a77为活动节点，a11后的垂直粗线段为分岔线，又称同步线，其后的多个活动可以并发执行，因此a22、a33和a44可以同时执行。图中的菱形条件为分支条件，用来表示满足某条件时执行某个活动。

【答案】B

【试题 10-18】面向对象技术中，组合关系表示(37)。(2012 年 5 月真题 37)

(37) A. 包与其中模型元素的关系　　B. 用例之间的一种关系

C. 类与其对象的关系　　D. 整体与其部分之间的一种关系

【解析】组合关系是关联关系的一种特例，体现的是一种 contains-a 的关系，这种关系比聚合更强，也称为强聚合；它同样体现整体与部分间的关系，但此时整体与部分是不可分的，整体的生命周期结束也就意味着部分的生命周期结束，比如人和大脑。组合跟聚合几乎相同，唯一的区别就是“部分”不能脱离“整体”单独存在，就是说，“部分”的生命期不能比“整体”还要长。

【答案】D

【试题 10-19】UML 序列图是一种交互图，描述了系统中对象之间传递消息的时间序列。其中，异步消息与同步消息不同，__(40)__。图 10-23 中__(41)__表示一条同步消息，__(42)__表示一条异步消息，__(43)__表示一条返回消息。(2012 年 5 月真题 40～43)

(40) A. 异步消息并不引起调用者终止执行而等待控制权的返回

B. 异步消息和阻塞调用有相同的效果

C. 异步消息是同步消息的响应

D. 异步消息和同步消息一样等待返回消息

(41) A. 1　B. 2　C. 3　D. 4

(42) A. 1　B. 2　C. 3　D. 4

(43) A. 1　B. 2　C. 3　D. 4

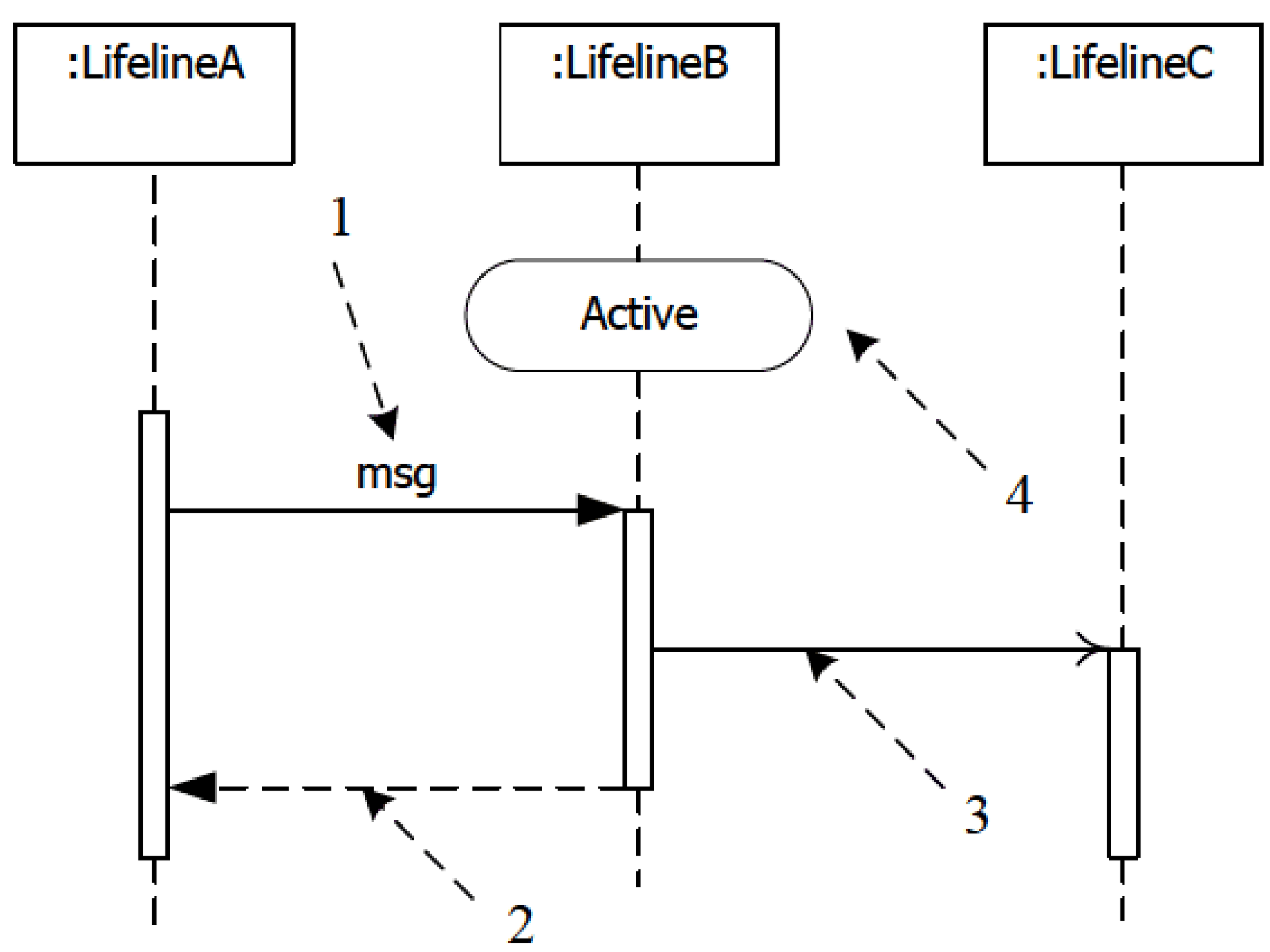


图 10-23　UML 序列图

【解析】序列图将交互关系表示为一个二维图。纵向是时间轴，时间沿竖线向下延伸。横向轴代表了在协作中各独立对象的类元角色。类元角色用生命线表示。当对象存在时，角色用一条虚线表示，当对象的过程处于激活状态时，生命线是一个双道线。消息用从一个对象的生命线到另一个对象生命线的箭头表示。箭头以时间顺序在图中从上到下排列。对于同步消息，发送方在继续之前，将等待同步消息响应。而对于异步消息，发送方在继续之前，无须等待响应的消息。

为了可读性，序列图的第一个消息总是从顶端开始，并且一般位于图的左边。然后继发的消息加入图中，稍微比前面的消息低些。为了显示一个对象(例如，生命线)传递一个消息给另外一个对象，画一条线指向接收对象，包括一个实心箭头(如果是一个同步调用操作)或一个棍形箭头(如果是一个异步信号)。消息/方法名字放置在带箭头的线上面。正在被传递给接收对象的消息，表示接收对象的类实现的一个操作/方法。返回消息是可选择的。一个返回消息画作一条带开放箭头的虚线，向后指向来源的生命线，在这条虚

线上面，放置操作的返回值。

【答案】(40)A；(41)A；(42)C；(43)B

【典型题 10-20】(44) 设计模式能使一个对象的状态发生改变时通知所有依赖它的监听者。(45) 设计模式限制类的实例对象只能有一个。适配器(Adapter)设计模式可以用于 (46) 。用于一个对象添加更多功能而不使用子类的是 (47) 设计模式。(2013 年 5 月真题 44～47)

(44) A. 责任链(Chain of Responsibility)　　B. 命令(Command)

C. 抽象工厂(Abstract Factory)　　D. 观察者(Observer)

(45) A. 原型(Prototype)　　B. 工厂方法(Factory Method)

C. 单例(Singleton)　　D. 生成器(Builder)

(46) A. 将已有类的接口转换成和目标接口兼容

B. 改进系统性能

C. 将客户端代码数据转换成目标接口期望的合适的格式

D. 使所有接口不兼容类可以一起工作

(47) A. 桥接(Bridge)　　B. 适配器(Adapter)

C. 组合(Composite)　　D. 装饰器(Decorator)

【解析】观察者模式(有时又被称为发布—订阅模式、模型—视图模式、源—收听者模式或从属者模式)是软件设计模式的一种。在此种模式中，一个目标对象管理所有相依于它的观察者对象，并且在它本身的状态改变时主动发出通知。这通常通过调用各观察者所提供的方法来实现。

单例模式是一种常用的软件设计模式。在它的核心结构中只包含一个被称为单例类的特殊类。通过单例模式可以保证系统中一个类只有一个实例而且该实例易于外界访问，从而方便对实例个数的控制并节约系统资源。

适配器模式(有时候也称包装样式或者包装)可以将一个类的接口适配成用户所期待的另外一个接口。一个适配允许通常因为接口不兼容而不能在一起工作的类工作在一起，做法是将类自己的接口包裹在一个已存在的类中。

装饰器模式就是创建一个新类为某一个类动态添加新功能或增强原有的功能。

【答案】(44)D；(45)C；(46)A；(47)D

【典型题 10-21】 (44) 设计模式定义一系列算法，把它们一个个封装起来，并且使它们可以相互替换。这一模式使得算法可独立于它的客户而变化。(2013 年 11 月真题 44)

(44) A. 策略(Strategy)　　B. 抽象工厂(Abstract Factory)

C. 观察者(Visitor)　　D. 状态(State)

【解析】策略模式定义了一系列的算法，并将每一个算法封装起来，而且还使它们可以相互替换。策略模式让算法独立于使用它的客户而变化；抽象工厂模式是所有形态的工厂模式中最为抽象和最具一般性的一种形态；观察者模式定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象。这个主题对象在状态发生变化时，会通知所有观察者对象，使它们能够自动更新自己。状态设计模式允许一个对象在其内部状态改变时改变它的行为。对象看起来似乎修改了它的类。所以本题答案为 A。

【答案】A

【典型题 10-22】在发布-订阅(Publish-Subscribe)消息模型中，订阅者订阅一个主题后，当该主题有新消息到达时，所有订阅者都会收到通知。 (45) 设计模式最适合这一模型。(2013 年 11 月真题 45)

(45) A. 适配器(Adapter)　　B. 通知(Notifier)
　　C. 状态(State)　　D. 观察者(Observer)

【解析】适配器设计模式是将一个类的接口转换成客户希望的另外一个接口；通知是一个对象对多个对象的同步操作；观察者模式定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象。这个主题对象在状态发生变化时，会通知所有观察者对象，使它们能够自动更新自己。状态设计模式允许一个对象在其内部状态改变时改变它的行为。

【答案】D

【典型题 10-23】图 10-24 所示为__(46)__设计模式，适用于：__(47)__。(2013 年 11 月真题 46～47)

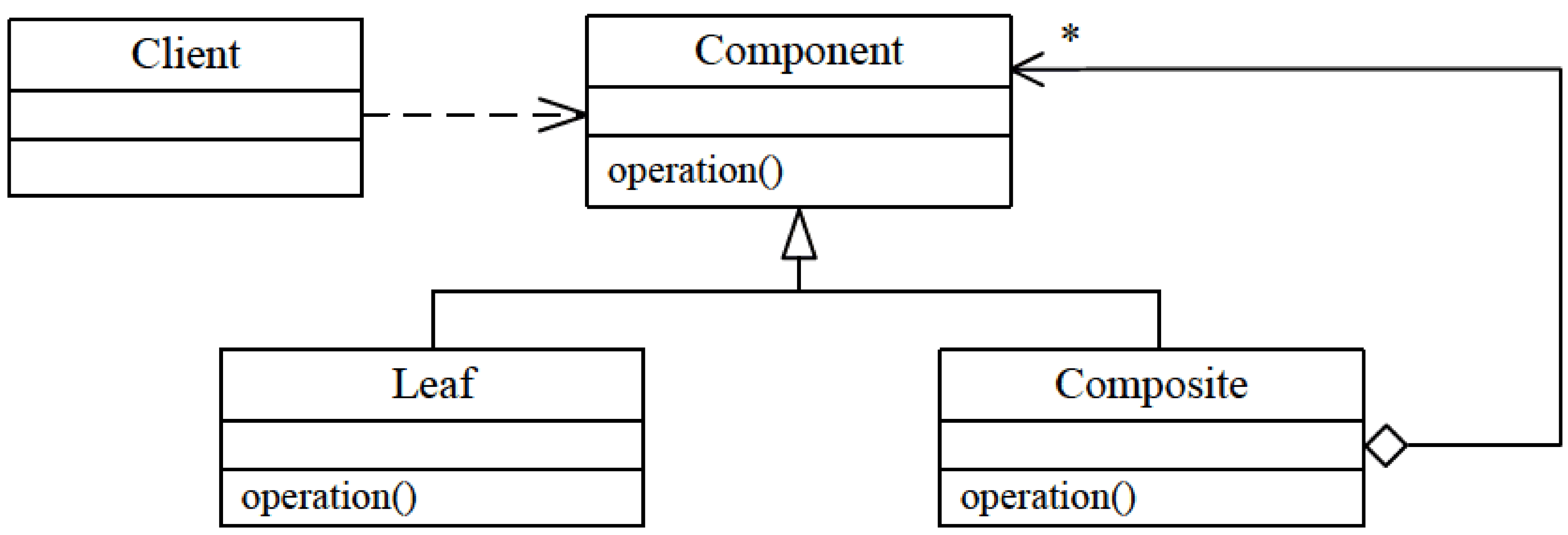


图 10-24　某设计模式的 UML 图

(46) A. 组件(Component)　　B. 适配器(Adapter)
　　C. 组合(Composite)　　D. 装饰器(Decorator)

(47) A. 表示对象的部分-整体层次结构
　　B. 不希望在抽象和它的实现部分之间有一个固定的绑定关系
　　C. 在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责
　　D. 使所有接口不兼容类可以一起工作

【解析】组合(Component)模式将对象组合成树形结构以表示“部分-整体”的层次结构，其中的组合对象使用户可以组合基元对象以及其他对象，从而形成任意复杂的结构。类 Component 为组合中的对象声明接口，类 Leaf 在组合中表示叶节点对象，并在组合中定义图元对象的行为。类 Composite 定义有子部件的那些部件的行为、存储子部件，并在 Component 接口中实现与子部件的有关操作。类 Client 通过 Component 接口操纵组合部件的对象。

【答案】(46)C；(47)A

【典型题 10-24】每种设计模式都有特定的意图。__(44)__模式使得一个对象在其内部状态改变时通过调用另一个类中的方法改变其行为，使这个对象看起来如同修改了它的类。图 10-25 是采用该模式的有关 TCP 连接的结构图实例。该模式的核心思想是引入抽象类__(45)__来表示 TCP 连接的状态，声明不同操作状态的公共接口，其子类实现与特定状态相关的行为。当一个__(46)__对象收到其他对象的请求时，它根据自身的当前状态做出不同的反应。(2012 年 11 月真题 44～46)

(44) A. 适配器(Adapter)　　B. 命令(Command)　　C. 观察者(Observer)　　D. 状态(State)

(45) A. TCPConnection　　B. state　　C. TCPState　　D. TCPEstablished

(46) A. TCPConnection　　B. state　　C. TCPState　　D. TCPEstablished

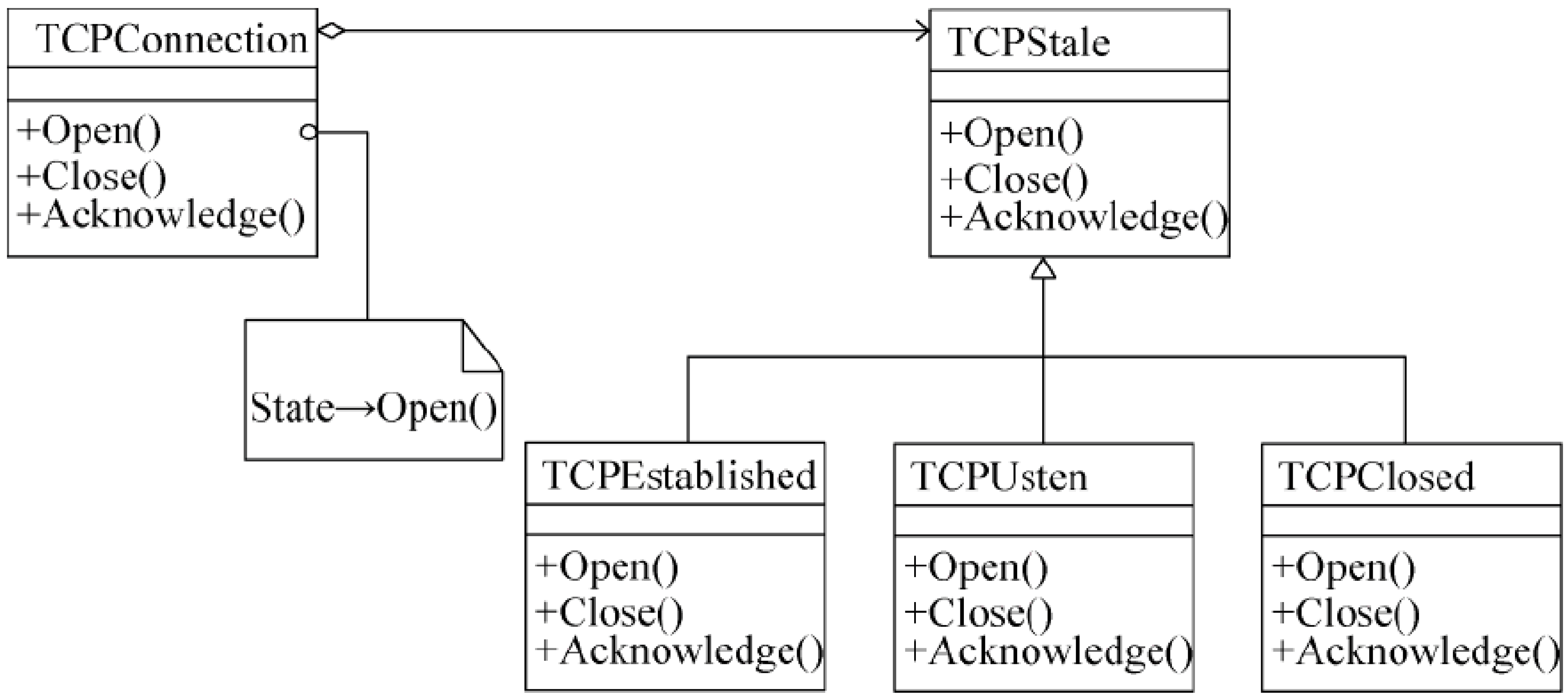


图 10-25　结构图

【解析】状态(State)模式允许一个对象在其内部状态改变时改变它的行为，使这个对象看起来如同修改了它的类。状态模式需要对每一个系统可能取得的状态创立一个状态类的子类，当系统的状态发生变化时，系统便改变所选的子类。在图 10-25 中，TCPConnection 为情景类，定义客户应用程序感兴趣的接口，维护一个具体状态(TCPEstablished、TCPListen 和 TCPClosed)子类的实例对象。TCPState 为抽象状态类，定义一个接口用来封装与 TCPConnection 的一个特别状态(State)相关的行为。TCPEstablished、TCPListen 和 TCPClosed 为具体类，每一个具体类实现一个 TCPConnection 的状态相关的行为。

【答案】(44)D；(45)C；(46)A

【典型题 10-25】欲使类 A 的所有使用者都使用 A 的同一个实例，应__(47)__。(2012 年 11 月真题 47)

(47) A. 将 A 标识为 final　　B. 将 A 标识为 abstract

C. 将单例(Singleton)模式应用于 A　　D. 将备忘(Memoto)模式应用于 A

【解析】单例(Singleton)模式用于确保某个类只有一个实例，且自行实例化，并向整个系统提供该实例。Singleton 模式为一个面向对象的应用程序提供了对象唯一的访问点，不管它实现何种功能，整个应用程序都会共享一个实例对象。

【答案】C

【典型题 10-26】设计模式根据目的进行分类，可以分为创建型、结构型和行为型三种。其中结构型模式用于处理类和对象的组合。__(44)__模式是一种结构型模式。(2012 年 5 月真题 44)

(44) A. 适配器(Adapter)　　B. 命令(Command)

C. 生成器(Builder)　　D. 状态(State)

【解析】结构型设计模式有：适配器(Adapter)模式、桥接器(Bridge)模式、组合(Component)模式、代理(Proxy)模式、享元(Flyweight)模式、Façade(外观)模式、装饰器(Decorator)模式。命令(Command)和状态(State)模式是行为型模式，生成器(Builder)是创建型模式。

【答案】A

【典型题 10-27】设计模式中的__(45)__模式将对象组合成树形结构以表示“部分-整体”的层次结构，使得客户对单个对象和组合对象的使用具有一致性。图 10-26 为该模式的类图，其中，__(46)__定义有子部件的那些部件的行为；组合部件的对象由__(47)__通过 component 提供的接口操作。(2012 年 5 月真题 45～47)

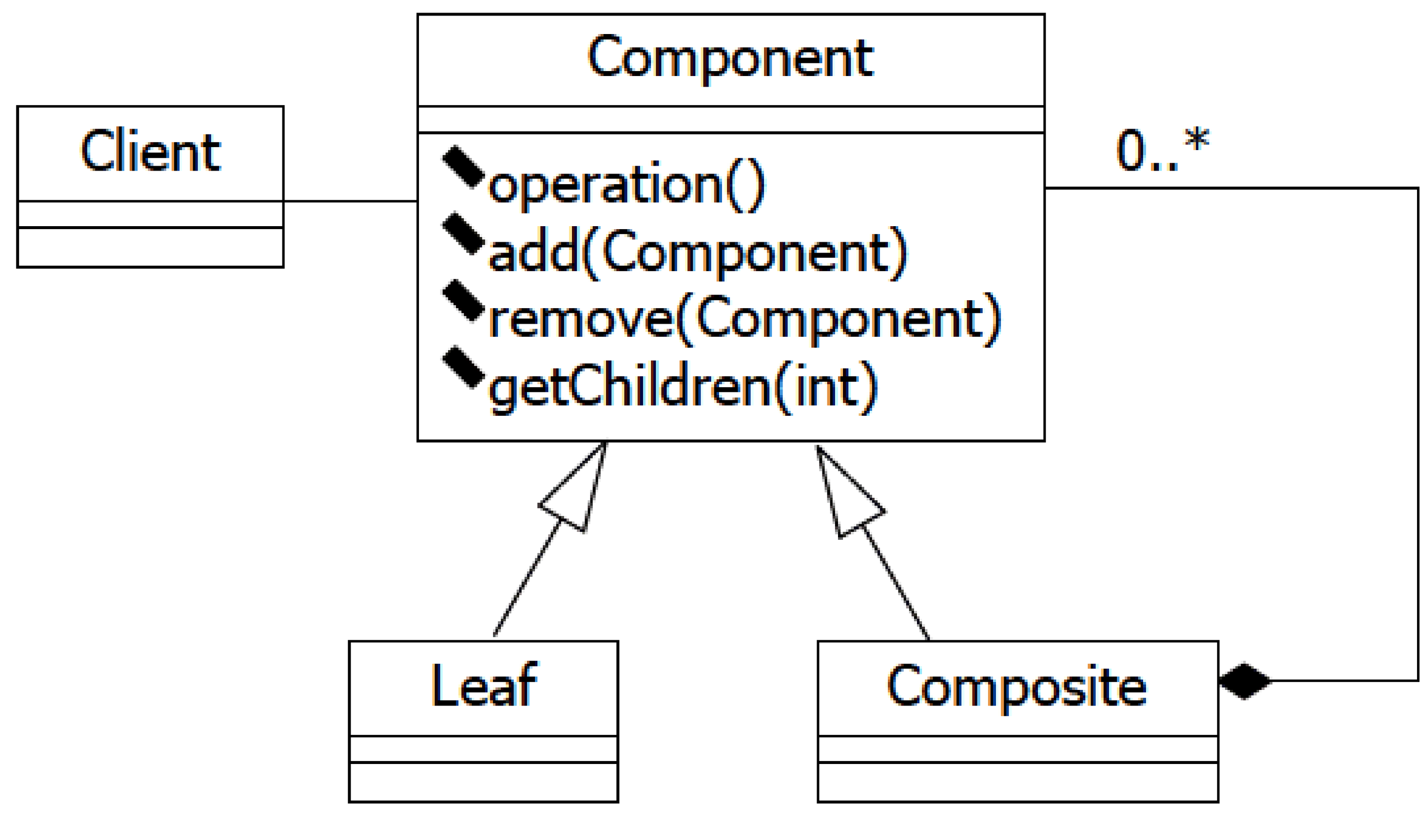


图 10-26

(45) A. 代理(Proxy)　　B. 桥接器(Bridge)

C. 组合(Composite)　　D. 装饰器(Decorator)

(46) A. Client　　B. Component　　C. Leaf　　D. Composite

(47) A. Client　　B. Component　　C. Leaf　　D. Composite

【解析】组合(Component)模式将对象组合成树形结构以表示“部分-整体”的层次结构，其中的组合对象使用户可以组合基元对象以及其他对象，从而形成任意复杂的结构。类Component为组合中的对象声明接口，类Leaf在组合中表示叶节点对象，并在组合中定义图元对象的行为。类Composite定义有子部件的那些部件的行为、存储子部件，并在Component接口中实现与子部件的有关操作。类Client通过Component接口操纵组合部件的对象。

【答案】(45)C；(46)D；(47)A

【典型题 10-28】(2014 年 5 月下午试题三)

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某高校图书馆欲建设一个图书馆管理系统，目前已经完成了需求分析阶段的工作，功能需求均使用用例进行描述，其中用例“借书(Check Out 800ks)”的详细描述如下。

参与者：读者<Patron>

典型事件流：

(1) 输入读者 ID；

(2) 确认该读者能够借阅图书，并记录读者 ID；

(3) 输入所要借阅的图书 ID；

(4) 根据图书目录中的图书 ID 确认该书可以借阅，计算归还时间，生成借阅记记录；

(5) 通知读者图书归还时间。

重复步骤(3)～(5)，直到读者结束借阅图书。

备选事件流：

2a.若读者不能借阅图书，说明读者违反了图书馆的借书制度(例如，没有支付借书费用等)。

①告知读者不能借阅，并说明拒绝借阅的原因；

②本用例结束。

4a.读者要借阅的书无法外借。

① 告知读者本书无法借阅；

② 回到步骤(3)。

说明：图书的归还时间与读者身份有关。如果读者是教师，图书可以借阅一年；如果是学生，则只能借阅 3 个月，读者 ID 中包含读者身份信息。

现采用面向对象方法开发该系统，得到如图 10-27 所示的系统类模型(部分)以及如图 10-28 所示的系统操作 CheckOut 的通信图(或协作图)。

【问题 1】

根据说明中的描述，以及图 10-27 和图 10-28，给出图 10-27 中 C1～C4 处所对应的类名(类名使用图 10-27 和图 10-28 中给出的英文词汇)。

【问题 2】

根据说明中的描述，以及图 10-27 和图 10-28，给出图 10-28 中 M1～M4 处所对应的方法名(方法名使用图 10-27 和图 10-28 中给出的英文词汇)。

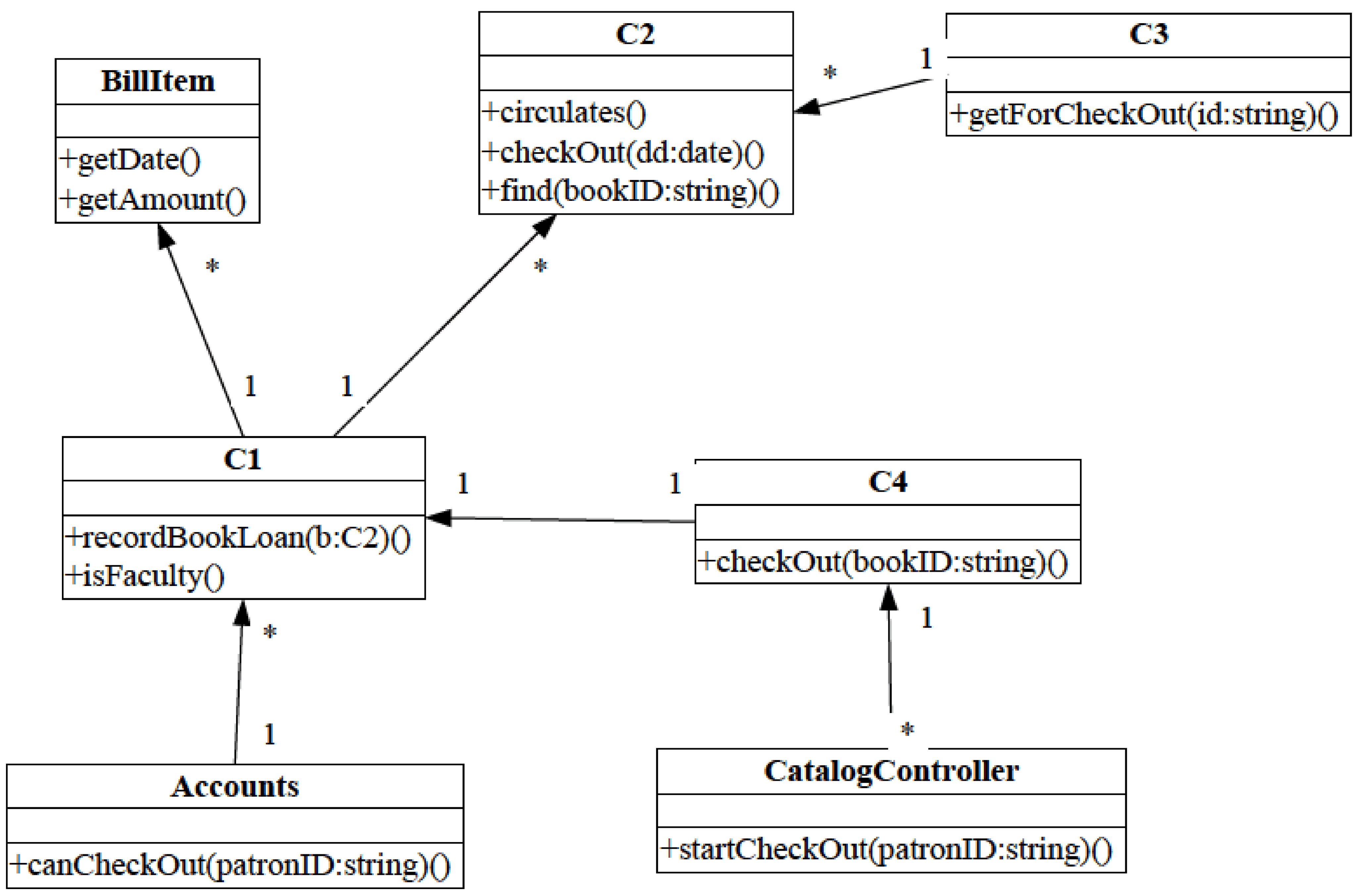


图 10-27　系统类模型

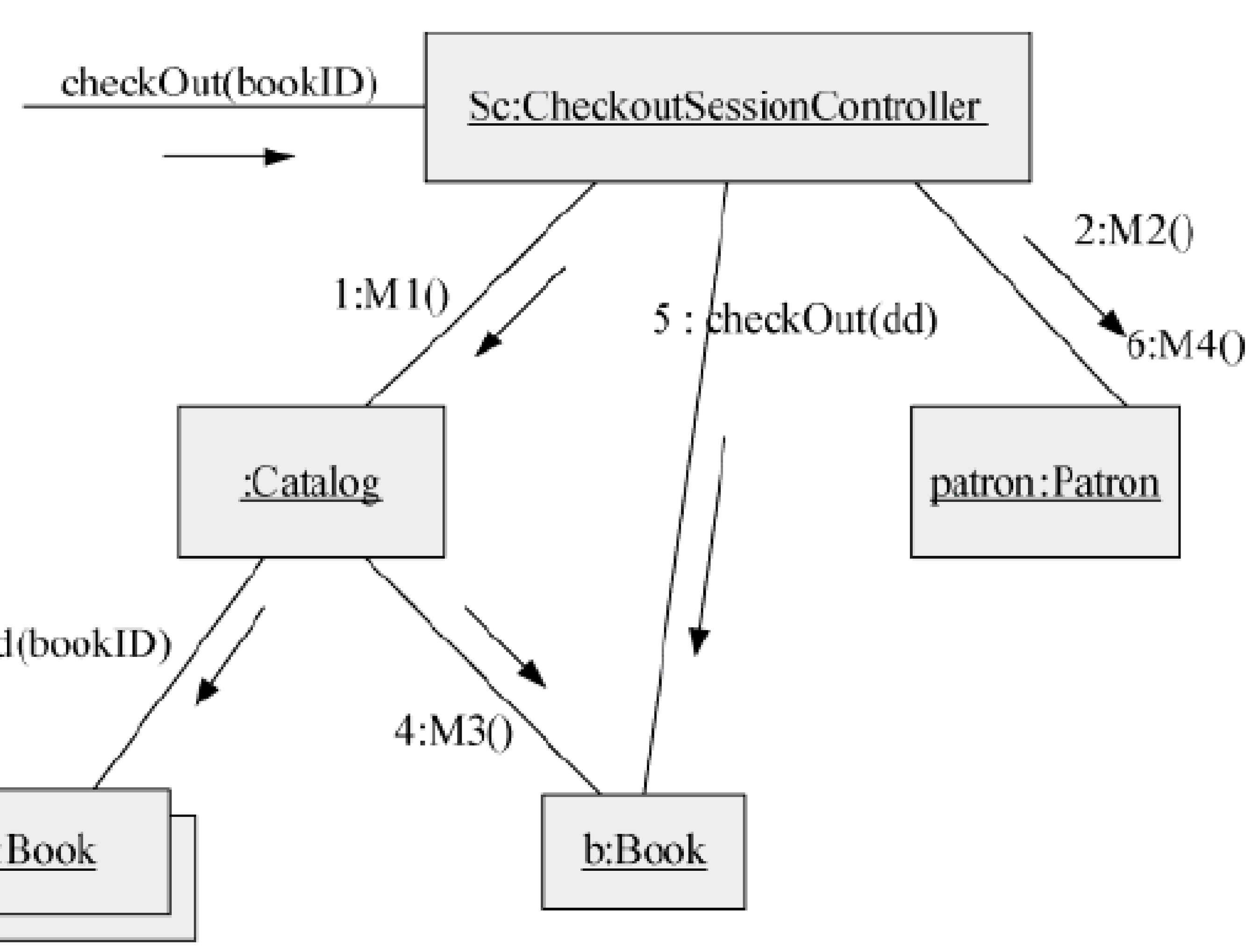


图 10-28　系统操作 checkout 的通信图

【问题 3】

用例“借书”的备选事件流 4a 中，根据借书制度来判定读者能否借阅图书，着图书馆的借书制度会不

断地扩充，并需要根据图书馆的实际运行情况来调整具体使用哪些制度，为满足这一要求，在原有类设计的基础上，可以采用何种设计模式？简要说明原因。

【解析】

【问题 1】 根据系统类模型，我们可以发现各个类之间的关联关系。

首先从类 Accouts 中的 canCheckOut(patronID:string)方法，可以看出 Accouts 关联 Patron，因此图中 C1 为 Patron。

C1 为 Patron，则 C1 必会与书关联，从 C1 中的 recordBookLoad(b,C2)，可以看出 C1 关联 C2，因此 C2 为 Book。

C2 为 Book，根据系统操作 checkOut 的通信图，可以看出与 Book 关联的是 Catalog，因此 C3 为 Catalog。

结合两图，则可以得出 C4 为 CheckoutSessioncontroller。

【问题 2】

结合典型事件流：

(1) 输入读者 ID。

(2) 确认该读者能够借阅图书，并记录读者 ID。

以上两步实际上就是判断读者是不是老师，也就是 isFaculty( )，因此 M2 为 isFaculty( )。

(3) 输入所要借阅的图书 ID；对应的操作就是 M1：getforcheck(bookID)。

(4) 根据图书目录中的图书 ID 确认该书可以借阅，计算归还时间，生成借阅记录；对应的操作就是 M3：circulates( )。

(5) 通知者图书归还时间，对应的操作就是 M4：recordBookLoan( )。

【答案】

【问题 1】

C1 : Patron　　C2 : Book　　C3 : Catalog　　C4 : CheckoutSessioncontroIler

【问题 2】

M1( ):getforcheck(bookID)；M2( ):isFacultyo；M3() : circulateso；M4() : recordBookLoan

【问题 3】

应采用策略模式，策略模式的优势在于可以灵活地添加对同一问题的不同处理方案，这与题目要求非常吻合。

【典型题 10-29】(2012 年 11 月下午试题三)

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某城市的各国家公园周边建造了许多供游客租用的小木屋和营地，为此，该城市设置了一个中心售票处和若干个区域售票处。游客若想租用小木屋或营地，必须前往中心售票处进行预定并用现金支付全额费用，所有的预定操作全部由售票处的工作人员手工完成。现欲开发一信息系统，实现小木屋和营地的预定及管理功能，以取代手工操作。该系统的主要功能描述如下：

(1) 管理预定申请。游客可以前往任何一个售票处提出预定申请，系统对来自各个售票处的预定申请进行统一管理。

(2) 预定。预定操作包含登记游客预定信息、计算租赁费用、付费等步骤。

(3) 支付管理。游客付费时可以选择现金和信用卡付款两种方式。使用信用卡支付可以享受 3%的折扣，

现金支付没有折扣。

(4) 游客取消预定。预定成功之后，游客可以在任何时间取消预定，但需支付赔偿金，剩余部分则退还给游客，赔偿金的计算规则是，在预定入住时间之前的 48 小时内取消，支付租赁费用 10%的赔偿金；在预定入住时间之后取消，则支付租赁费用 50%的赔偿金。

(5) 自动取消预定。如果遇到恶劣天气(如暴雨、山洪等)，系统会自动取消所有的预定，发布取消预定消息，全额退款。

(6) 信息查询。售票处工作人员查询小木屋和营地的预定情况和使用情况，以判断是否能够批准游客的预定申请。

现采用面向对象方法开发上述系统，得到如表 10-7 所示的用例列表和表 10-8 所示的类列表，对应的用例图和类图分别如图 10-29 和图 10-30 所示。

表 10-7　用例列表

用例名	说　明	用例名	说明
ManageInquiries	管理预定申请	ManageCashPayment	现金支付
MakeReservation	预定	ManageCrCardPayment	计算付款折扣
ManagePayment	支付管理	GetDiscount	系统自动取消预定
CancelReservation	游客取消预定	AutoCancelReservation	计算取消预定的赔偿金
CheckAvailability	信息查询	CalculateRefund	
PublishMessage	发布取消预定消息		

表 10-8　类列表

类　名	说　明	类　名	说　明
NationalPark	国家公园	Customer	游客
Reservation	预定申请	ReservationItem	预定申请内容
TicketingOfficer	售票处	CampSite	营地
Bungalow	小木屋	Payment	付款
Discount	付款折扣	CashPayment	现金支付
CreditCardPayment	信用卡支付	Rate	租凭费用

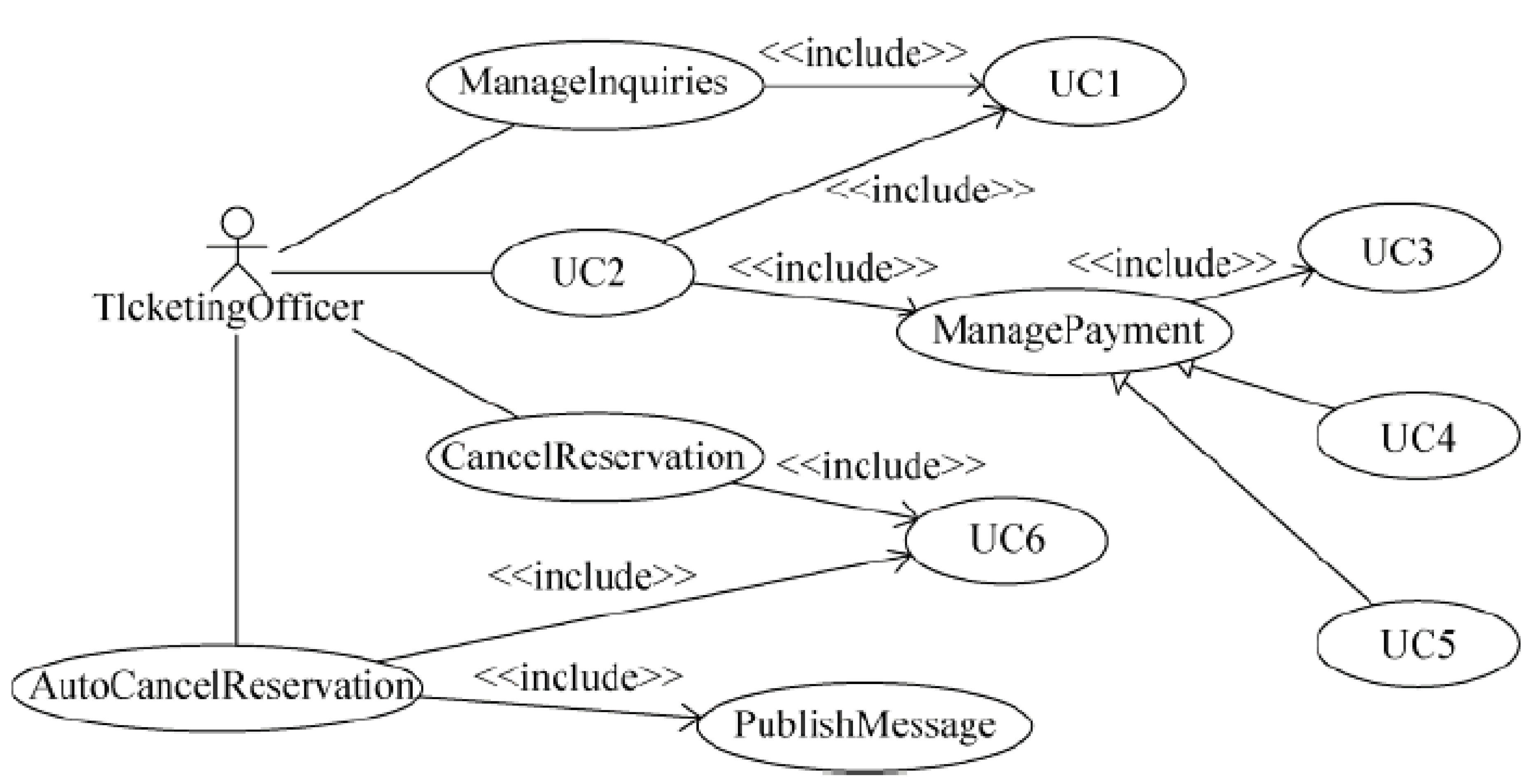


图 10-29　用例图

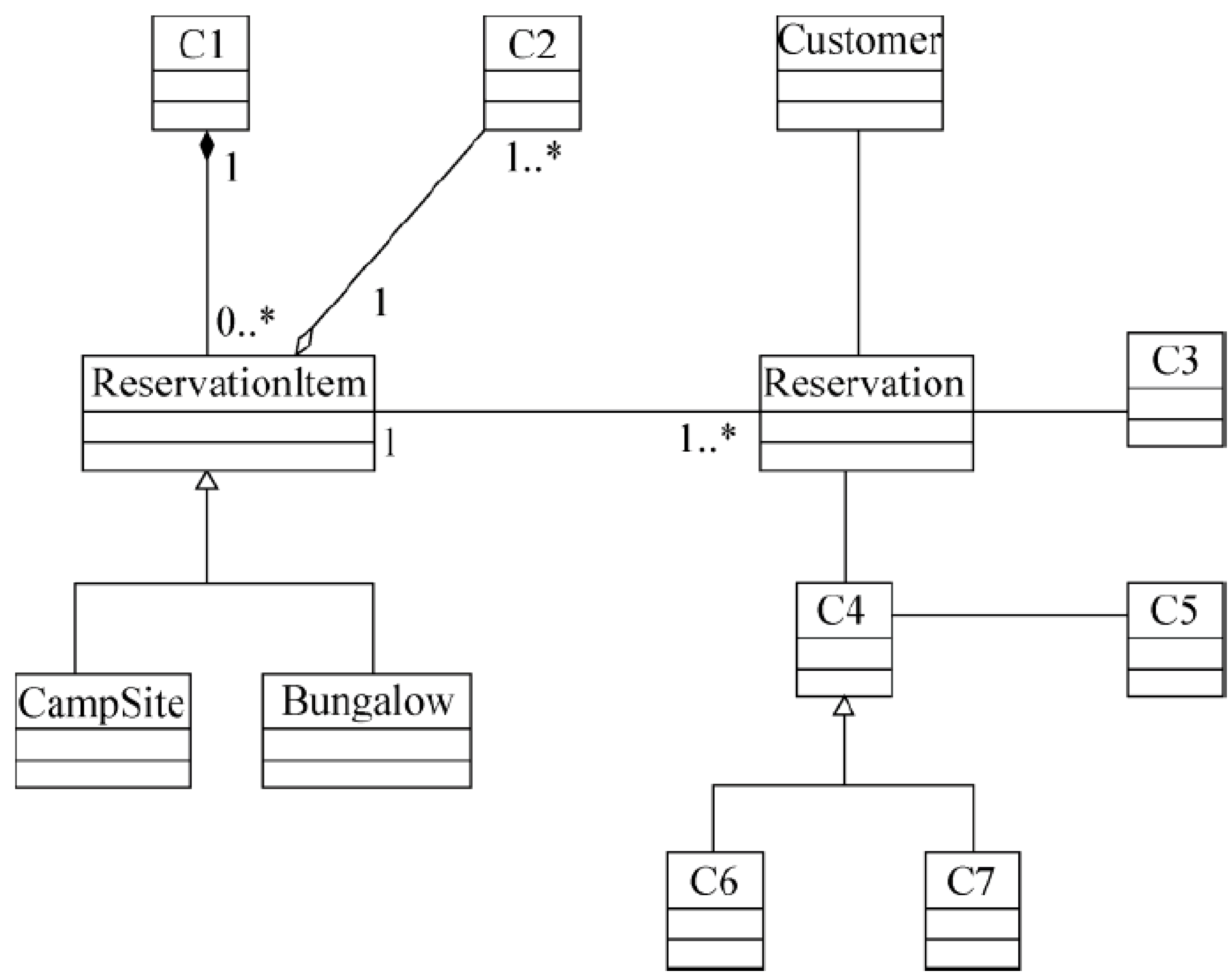


图 10-30　类图

【问题 1】

根据说明中的描述与表 10-7，给出图 10-29 中 UC1～UC6 处所对应的用例名称。

【问题 2】

根据说明中的描述与表 10-8，给出图 10-30 中 C1～C7 处所对应的类名。

【问题 3】

对于某些需求量非常大的小木屋或营地，说明中功能(4)的赔偿金计算规则，不足以弥补取消预定所带来的损失，如果要根据预定的时段以及所预定场地的需求量，设计不同层次的赔偿金计算规则，需要对图 10-30 进行怎样的修改？(请用文字说明)

【解析】

本题主要考查在面向对象分析与设计过程中，如何利用用例图和类图描述系统需求模型及设计模型。考生需要理解面向对象方法的相关概念和思想，并熟悉 UML 的语法及应用。用例图和类图是考题中出现次数最多的两种 UML 模型。

【问题 1】

本问题要求将图 10-29 中的用例名称补充完整。用例图的构成要素有：参与者、用例以及用例之间的关系。首先找出本题的突破口，即图 10-29 中，ManagePayment 与 UC4、UC5 间的泛化关系。由“游客付费时可以选择现金和信用卡付款两种方式”，可知 UC4 和 UC5 为 ManageCashPayment 及 ManageCrCardPayment。ManagePayment 与 UC3 之间为<<include>>关系，又由“信用卡支付可以享受折扣，现金支付没有折扣”，可知 UC3 为 GetDiscount。由 CancelReservation 和 AutoCancelReservation 用例与 UC6 之间的关系，取消预定后需要支付赔偿金，因此 UC6 为 CalculateRefund。此外，ManageInquiries 是管理预定请求用例，在预定时包含登记游客信息、计算租赁费用等，因此 UC1 为 CheckAvailablity，UC2 为 MakeReservation。

【问题 2】

本问题考查的是类图建模，解题的重点在于根据类图中提供的类及类之间的关联关系，推断出剩余的类。从图 10-30 可以看出，需要补充的类集中在三个结构：聚集结构(C1 和 C2)、泛化结构(C4、C6 和 C7)

和关联结构(C3 和 C5)。泛化结构是比较容易辨识的类之间的关联关系，是一种特殊与一般关系。由说明中 “游客付费时可以选择现金和信用卡付款两种方式”及“信用卡支付可以享受折扣，现金支付没有折扣” 的描述，可以确定 C4 为 Payment，C5 为 Discount，C6 为 CashPayment，C7 为 CreditCardPayment，C6 和 C7 可以互换。由于 ReservationItem 与 C1、C2 间是聚集结构，其之间存在“部分/整体”关系，并由其之间的多重度可知，C1 为 NationPark，C2 为 Rate。最后，结合表 10-8 以及 Reservation 与 C3 间的关系，可判断 C3 为 Ticketing officer。

【问题 3】

本问题较为新颖，在以往考试中出现较少。

题目要求根据预定的时段以及所预定场地的需求量，设计不同层次的赔偿金计算规则。在图 10-30 中，和赔偿金相关的类是 ReservationItem 和 Rate。因此可以通过新增加一个类，使该类与类 ReservationItem 之间有关联关系，或直接修改 Rate 类，使其具有计算不同层次赔偿金的功能。

【答案】

【问题 1】

UC1：CheckAvailablity	UC2：　MakeReservation	UC3：　GetDiscount
UC4：ManageCashPayment	UC5：　ManageCrCardPayment	UC6：　CalculateRefund

【问题 2】

C1：NationPark	C2：Rate	C3：Ticketing officer
C4：Payment	C5：Discount	C6：CashPayment
C7：CreditCardPayment		

【问题 3】

修改方案 1：增加一个新类，该类与类 ReservationItem 之间有关联关系。

修改方案 2：修改 Rate 类，使其具有计算赔偿金的功能。

【典型题 10-30】(2012 年 5 月下午试题三)

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某网上购物平台的主要功能如下。

(1) 创建订单。顾客(Customer)在线创建订单(Order)，主要操作是向订单中添加项目、从订单中删除项目。订单中应列出所订购的商品(Product)及其数量(quantities)。

(2) 提交订单。订单通过网络来提交。在提交订单时，顾客需要提供其姓名(name)、收货地址(address)以及付款方式(form of payment)(预付卡、信用卡或者现金)。为了制订送货计划以及安排送货车辆，系统必须确定订单量(volume)。除此之外，还必须记录每种商品的名称(name)、造价(cost price)、售价(sale price)以及单件商品的包装体积(cubic volume)。

(3) 处理订单。订单处理人员接收来自系统的订单，根据订单内容，安排配货，制订送货计划。在送货计划中不仅要指明发货日期(delivery date)，还要记录每个订单的限时发送要求(Delivery Time Window)。

(4) 派单。订单处理人员将已配好货的订单转交给派送人员。

(5) 送货/收货。派送人员将货物送到顾客指定的收货地址。当顾客收货时，需要在运货单(delivery slip)上签收。签收后的运货单最终需交还给订单处理人员。

(6) 收货确认。当订单处理人员收到签收过的运货单后，会和顾客进行一次再确认。

现采用面向对象方法开发上述系统，得到如图10-31所示的用例图和图10-32所示的类图。

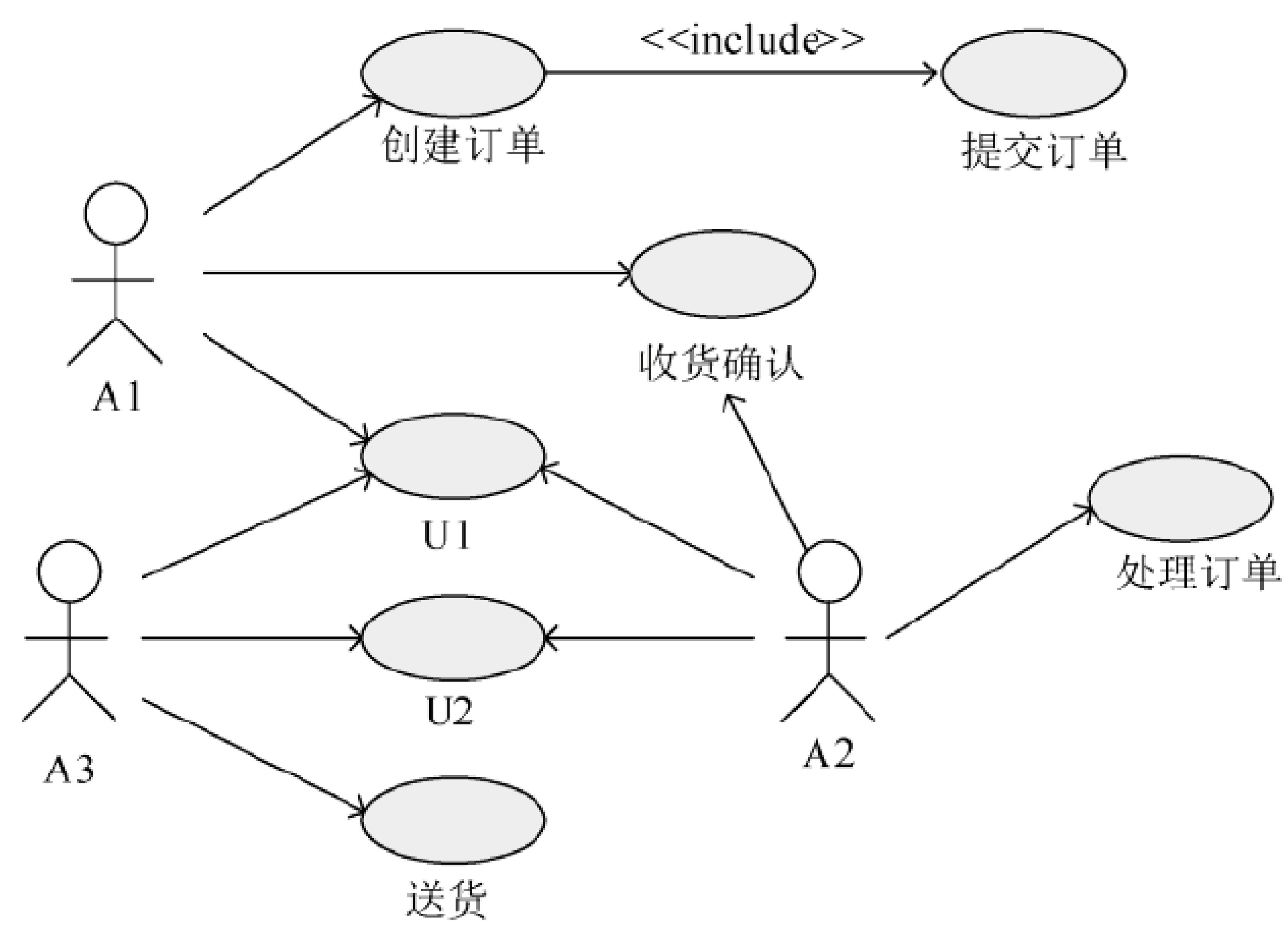


图 10-31　用例图

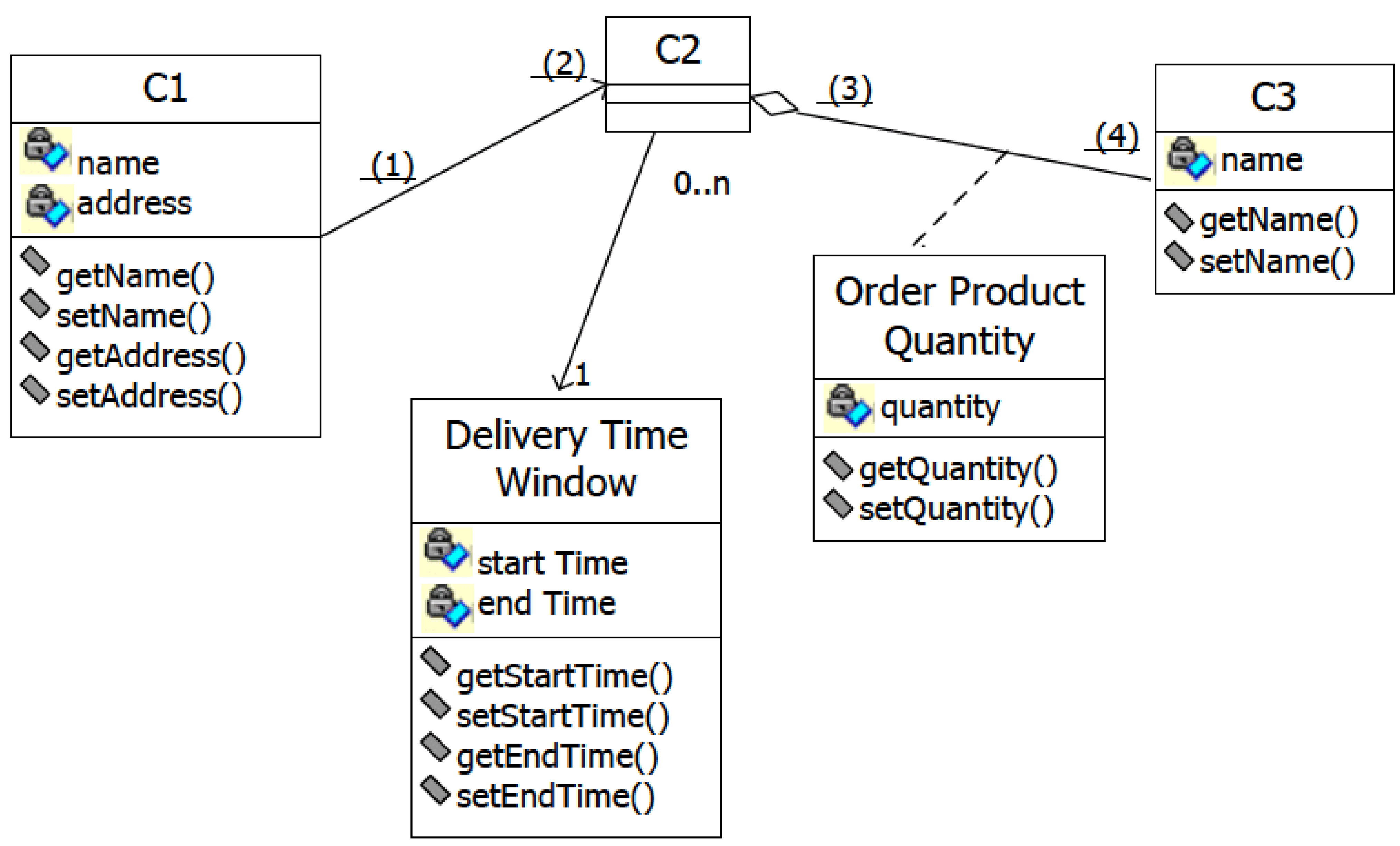


图 10-32　类图

【问题 1】

根据说明中的描述，给出图 13-31 中 A1～A3 所对应的参与者名称和 U1～U2 处所对应的用例名称。

【问题 2】

根据说明中的描述，给出图 10-32 中 C1～C3 所对应的类名以及(1)～(4)处所对应的多重度(类名使用说明中给出的英文词汇)。

【问题 3】

根据说明中的描述，将类 C2 和 C3 的属性补充完整(属性名使用说明中给出的英文词汇)。

【解析】

【问题 1】由“顾客(Customer)在线创建订单(Order)”可知，A1 为顾客。由“订单处理人员接收来自系统的订单”、“当订单处理人员收到签收过的运货单后，会和顾客进行一次再确认”可知，A2 为订单处理人员。A3 负责送货，很显然，A3 为派送人员。顾客创建好订单后，提交订单，由订单处理人员配货，然后进行派单，将已配好货的订单转交给派送人员，可见 U2 处的用例为“派单”。然后派送人员送货，

顾客收货，因此 U1 为“收货”。

【问题 2】类的 UML 表示是一个长方形，垂直地分为三个区。顶部区域显示类的名字。中间的区域列出类的属性。底部的区域列出类的操作。由题目提供的类图可知，C1 有 name 属性和 address 属性；C3 有 name 属性。C1 单向关联 C2，C1 可以调用 C2 的公共属性和方法。C2 和 C3 之间是聚合的关系，表示 C2 聚合 C3，但是 C3 可以离开 C2 而独立存在。在 C2 和 C3 类之间的关联，产生了称为 Order Product Quantity 的关联类。C2 是整个类图的中心，购物的所有活动都是围绕着订单展开的，C2 为 Order。订单中记录着所订购的商品，C3 为 Product。用户创建订单，需要调用订单的属性和方法，C1 为 Customer。

UML 中关联的多重度是指一个类的实例能够与另一个类的多少个实例相关联。具体取值的含义有：

- 0..1：0 个或 1 个
- 1：只能 1 个
- 0..*：0 个或多个
- *：0 个或多个
- 1..*：1 个或多个

一个客户可以提交多份订单，一份订单只能是某一个客户的，当然，客户也可以不提交订单，当一个 Customer 实体存在时，可以有也可以没有 Order 实体与之关联；而当一个 Order 实体存在时，必然有一个 Customer 实体与之关联。显然(1)处多重度为 1，(2)处为 0..*。(3)和(4)处要填入的是订单和商品实体之间的关联，订单中至少有一件商品，因此(4)处应填入 1..*；商品和订单之间是聚合的关系，商品可以包含在订单中，商品也可以独立存在，不属于任何订单，因此(3)处应填入 0..*。

【问题 3】C2 是订单类，订单中应列出所订购的商品(Product)及其数量(quantities)。在提交订单时，顾客需要提供其姓名(name)、收货地址(address)以及付款方式(form of payment)。为了制订送货计划以及安排送货车辆，系统必须确定订单量(volume)。Order 类聚合 Product 类，Product 的属性中有商品的相关信息，关联类 Order Product Quantity 中有 quantity 属性，因此 Order 类中需要有付款方式(form of payment)属性、订单量(volume)属性。在送货计划中要指明发货日期(delivery date)，因此 Order 类中还要有发货日期(delivery date)属性。

C3 是商品类，商品的属性有名称(name)、造价(cost price)、售价(sale price)以及单件商品的包装体积(cubic volume)。

【答案】

【问题 1】A1：顾客　　A2：订单处理人员　　A3：派送人员　　U1：收货　　U2：派单

【问题 2】C1：Customer　C2：Order　　C3：Product

(1)1　　(2)0..n 或 0..*　　(3)0..n 或 0..*　　(4)1..n 或 1..*

【问题 3】C2：volume、delivery date、form of payment

C3：cubic volume、cost price、sale price

【典型题 10-31】(2013 年 5 月下午试题五)

阅读下列说明和 C++代码，将应填入<u>(*n*)</u>处的字句写在答题纸的对应栏内。

【说明】

现要求实现一个能够自动生成求职简历的程序，简历的基本内容包括求职者的姓名、性别、年龄及工作经历。希望每份简历中的工作经历有所不同，并尽量减少程序中的重复代码。

现采用原型模式(Prototype)来实现上述要求，得到如图 10-33 所示的类图。

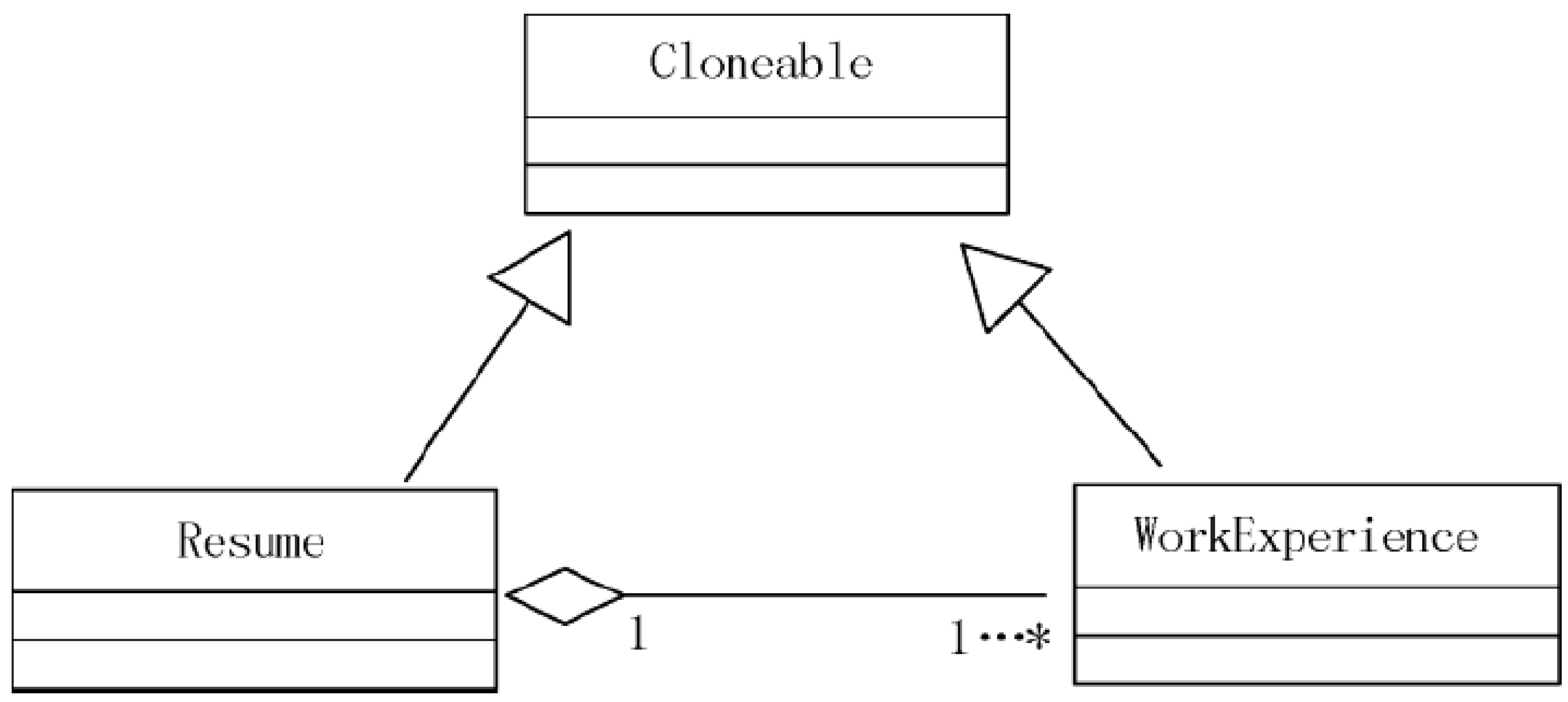


图 10-33　类图

【C++代码】

```
#include<string>
Using namespace std;
class Cloneable{
public:
     ______(1)______;
};
class WorkExperience:public Cloneable{    //工作经历
private:
      String  workDate;
      String  company;
public:
   Cloneable*Clone(){
   _________(2)_________;
   Obj->workDate=this->workDate;
   Obj->company=this->company;
   Return obj;
   }
     //其余代码省略
};
class Resume:public Cloneable{  //简历
private:
   string  name;  string sex;  string  age;
   WorkExperience* work;
   Resume(WorkExperience*work){
      This->work=____(3)______;
   }
public:
     Resume(string name){  /* 实现省略*/    }
     void  SetPersonalInfo(string sex,string age){     /* 实现省略*/      }
     void setWorkExperience(string  workDate,string company){  /* 实现省略
*/   }
     Cloneable*Clone(){
      ____________(4)______;
         Obj->name=this->name;
         Obj->sex=this->sex;
         Obj->age=this->age;
```

```
            return obj;
        }
    };
    int main(){
        Resume *a=new Resume("张三");
        A->SetPersonalInfo("男", "29");
        A->SetWorkExperience("1998～2000", "XXX 公司");
        Resume*b=______(5)______;
        B->SetWorkExperience("2001～2006", "YYY 公司");
        Return 0;
    }
```

【解析】

本题考查原型模式的概念及应用，原型模型是一种对象创建模型，用原型实体指定创建对象的种类，并且通过复制这些原型创建新的对象。原型模型允许一个对象再创建另一个可订制的对象，无须知道任何创建的细节。

Prototype 模式其实就是常说的“虚拟构造函数”一个实现，C++的实现机制中并不支持这个特性，但是通过不同派生类实现的 Clone 接口函数可以完成与“虚拟构造函数”同样的效果。

题中声明一个虚拟基类，所有的原型都从这个基类继承，(1)所代表的就是这个基类中的纯虚函数，需要供继承者自行实现，即为 virtual Cloneable * Clone() = 0，首先声明一个抽象基类，并定义 clone()函数为纯虚函数。然后根据基类实例化各个子类，并且实现复制构造函数，并实现 clone()函数，由此可知(2)处为 WorkExperience *obj，(3)处为 Work，(4)处为 Resume *obj。在 main 函数中实现 Resume *b 对*a 的复制，故根据 C++语法，(5)中为 a->Clone( )。

【答案】

(1)　virtual Cloneable * Clone() = 0　　(2)　WorkExperience *obj

(3)　Work　　(4)　Resume *obj　　(5)a->Clone( )

【典型题 10-32】(2012 年 11 月下午试题五)

阅读下列说明和 C++代码，将应填入<u>(n)</u>处的字句写在答题纸的对应栏内。

【说明】

现欲开发一个软件系统，要求能够同时支持多种不同的数据库，为此采用抽象工厂模式设计该系统。以 SQL Server 和 Access 两种数据库以及系统中的数据库表 Department 为例，其类图如图 10-34 所示。

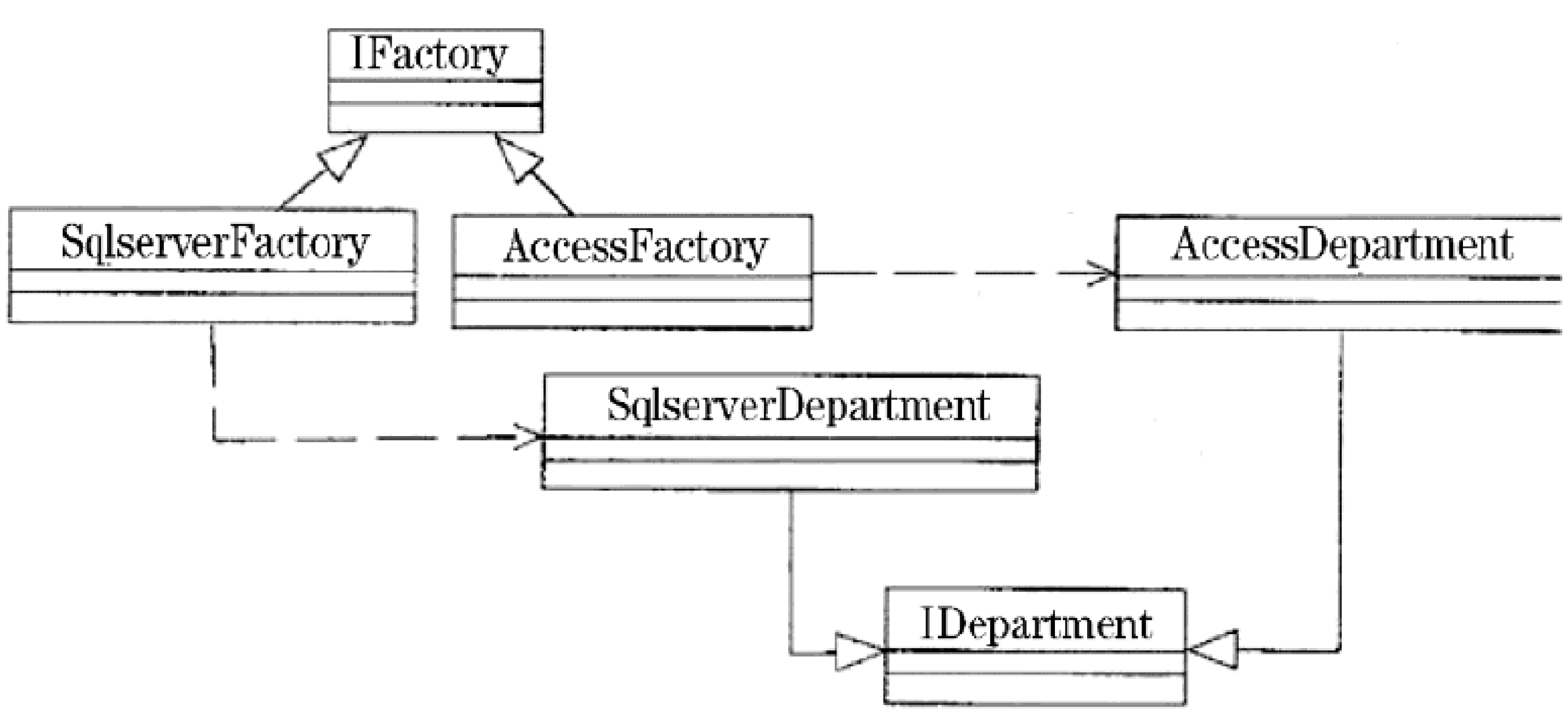


图 10-34　类图

【C++代码】

```
#include <iostream>
using namespace std;
class Department{/*代码省略*/};
class IDepartment{
public:
    ___(1)___=0;
    ___(2)___=0;
}
class SqlserverDepartment:___(3)___{
public:
    void Insert(Department* department){
        cout<<"Insert a record into Department in SQL Server!\n";
        //其余代码省略
    }
    Department GetDepartment(int id){
        /*代码省略*/
    }
};
class AccessDepartment:___(4)___{
public:
    void Insert(Department* department){
        cout<<"Insert a record into Department in ACCESS!\n";
        //其余代码省略
    }
    Department GetDepartment(int id){
        /*代码省略*/
    }
};
___(5)___{
public:
    ___(6)___=0;
}
class SqlServerFactory:public IFactory{
public:
    IDepartment* CreateDepartment(){ return new SqlserverDepartment(); }
    //其余代码省略
};
class AccessFactory:public IFactory{
public:
    IDepartment* CreateDepartment(){ return new AccessDepartment(); }
    //其余代码省略
};
```

【解析】

本题考查抽象工厂设计模式的概念及其应用。

抽象工厂设计模式的意图是：提供一个创建一系列相关或相互依赖的对象，而无须指出它们具体的类。在如下情况下应当考虑使用抽象工厂模式：

- 当一个系统要独立于它的产品的创建、组合和表示时；
- 当一个系统要由多个产品系列中的一个来配置时；
- 当需强调一系列相关的产品对象的设计以便进行联合使用时；
- 想提供一组对象而不显示它们的实现过程，只显示它们的接口时。

抽象工厂设计模式的类图如图 10-35 所示，其中：

- AbstractFactory 为抽象工厂，声明抽象产品的方法。
- ConcreteFactory 为具体工厂，执行生成抽象产品的方法，生成一个具体的产品。
- AbstractProduct 为抽象产品，为一种产品声明接口。
- Product 为具体产品，定义具体工厂生成的具体产品的对象，实现产品接口。

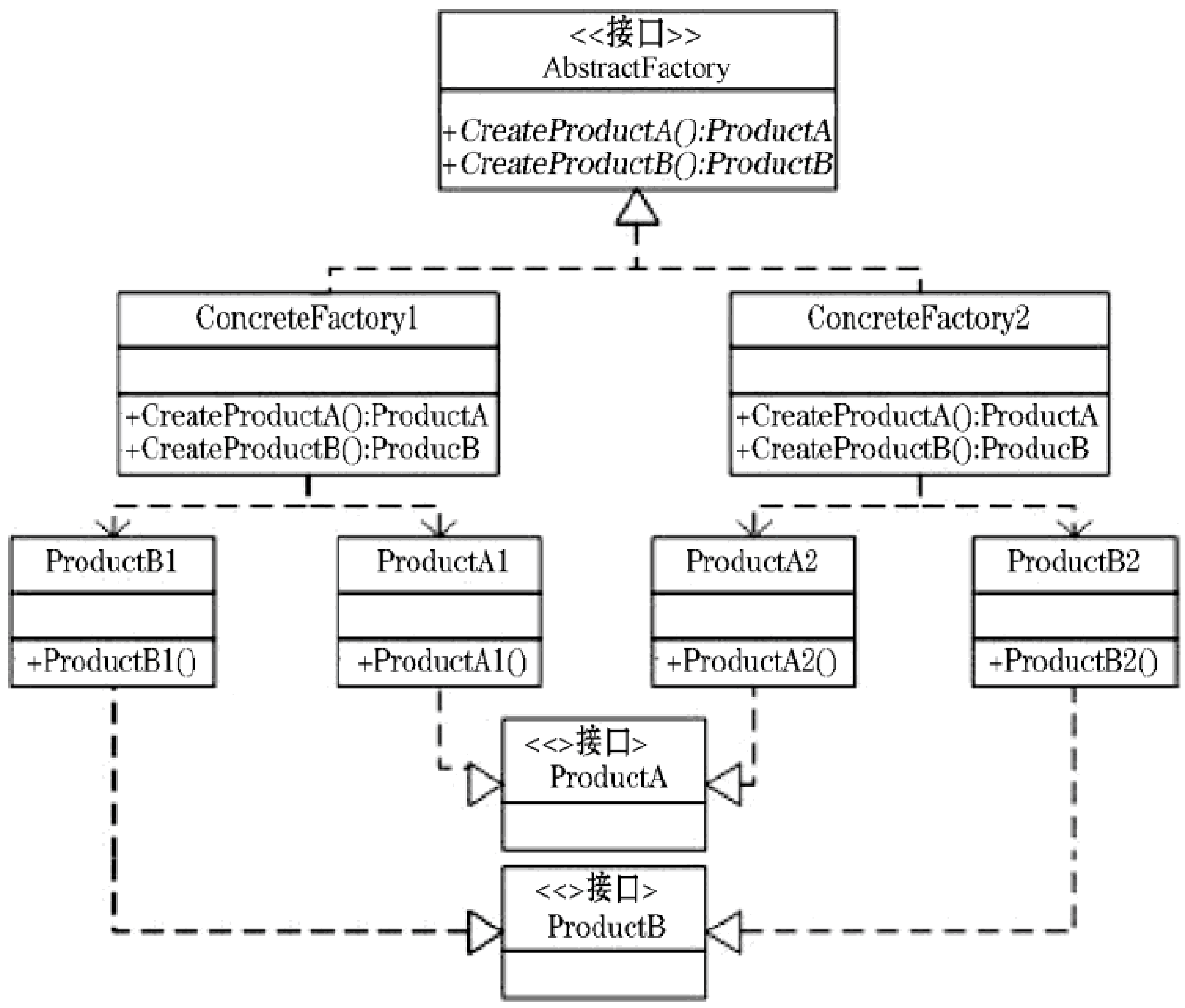


图 10-35　抽象工厂设计模式类图

图 10-34 中的 IFactory 对应类图中的 AbstractFactory，Sqlserverfactory 和 Accessfactory 对应类图中的 ConcreteFactory，SqlserverDepartment 和 AccessDepartment 对应类图中的 Product，而 IDepartment 对应类图中的 AbstractProduct。

由于类 IDepartment 的作用是为其子类提供接口，所以将其定义为抽象类。在 C++中，抽象类中至少包含一个纯虚函数的类，而纯虚函数是没有函数体的函数，其作用是为其子类提供统一的接口。若要使用纯虚函数，必须在子类中进行重置。空(1)和空(2)考查的是如何定义抽象类 IDepartment。从 IDepartment 的子类的方法中可以确定空(1)处应填入 virtual void Insert(Department* department)，空(2)处应填入 virtual Department GetDepartment(int id)。

空(3)和空(4)考查继承的概念和语法。由于 SqlserverDepartment 和 AccessDepartment 均继承 IDepartment，因此，空(3)和空(4)都应填入 public IDepartment。

由于所给程序中缺少有关抽象类 IFactory 的定义，因此空(5)处应为定义类 IFactory，应填入 class IFactory。抽象类 IFactory 中至少需包含一个纯虚函数的类，由其子类 Sqlserverfactory 和 Accessfactory 中

方法的定义可知，(6)处应填入 virtual IDepartment* CreateDepartment()。

【答案】

(1) virtual void Insert(Department* department)

(2) virtual Department GetDepartment(int id)

(3) public IDepartment

(4) public IDepartment

(5) class IFactory

(6) virtual IDepartment* CreateDepartment()

【典型题 10-33】(2012 年 5 月下午试题五)

阅读下列说明和 C++代码，将应填入(n)处的字句写在答题纸的对应栏内。

【说明】

某咖啡店在卖咖啡时，可以根据顾客的要求在其中加入各种配料，咖啡店会根据所加入的配料来计算费用。咖啡店所供应的咖啡及配料的种类和价格如表 10-9 所示。

表 10-9 咖啡及配料的种类和价格

咖 啡	价格/杯(￥)	配 料	价格/杯(￥)
蒸馏咖啡(Espresso)	25	摩卡(Mocha)	10
深度烘焙咖啡(DarkRoast)	20	奶泡(Whip)	8

现采用装饰器(Decorator)模式来实现计算费用的功能，得到如图 10-36 所示的类图。

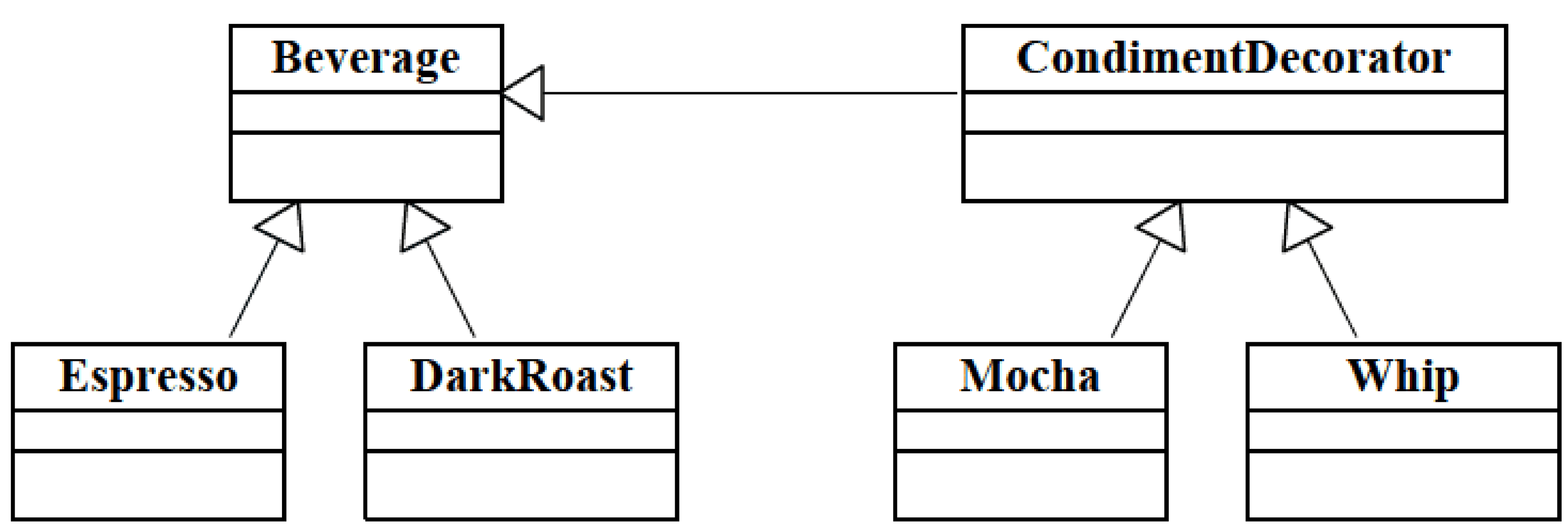


图 10-36 类图

【C++代码】

```
#include <iostream>
#include <string>
using namespace std;
const int ESPRESSO_PRICE = 25;
const int DRAKROAST_PRICE = 20;
const int MOCHA_PRICE = 10;
const int WHIP_PRICE = 8;
class Beverage { //饮料
    (1) : string description;
public:
    (2) (){ return description; }
```

```
        (3)  ;
};
class CondimentDecorator : public Beverage { //配料
protected:
        (4) ;
};
class Espresso : public Beverage { //蒸馏咖啡
public:
        Espresso () {description="Espresso"; }
        int cost(){return ESPRESSO_PRICE; }
};
class DarkRoast : public Beverage { //深度烘焙咖啡
public:
        DarkRoast(){ description = "DardRoast"; }
        int cost(){ return DRAKROAST_PRICE; }
};
class Mocha : public CondimentDecorator { //摩卡
public:
        Mocha(Beverage*beverage){ this->beverage=beverage; }
        string getDescription(){ return beverage->getDescription()+",Mocha"; }
        int cost(){ return MOCHA_PRICE+beverage->cost(); }
};
class Whip :public CondimentDecorator { //奶泡
public:
        Whip(Beverage*beverage) { this->beverage=beverage; }
        string getDescription() {return beverage->getDescription()+",Whip"; }
        int cost() { return WHIP_PRICE+beverage->cost(); }
};
int main() {
        Beverage* beverage = new DarkRoast();
        beverage=new Mocha( (5) );
        beverage=new Whip( (6) );
        cout<<beverage->getDescription()<<"￥"<<beverage->cost()<<endl;
        return 0;
}
```

编译运行上述程序，其输出结果为：

```
DarkRoast, Mocha, Whip ,￥38
```

【解析】由类图可知，Beverage 是基类，Espresso、DarkRoast、CondimentDecorator 是 Beverage 的派生类，Mocha、Whip 又是 CondimentDecorator 的派生类。

(1) 处应填入 description 的访问控制类型，可能为 private 或 protected。在 Beverage 的派生类 Espresso 的初始化函数中直接使用了 description，由此可知，在基类中，description 的访问控制类型为 protected。如果为 private，则在派生中不能使用。

在基类中先后动态建立了一个 DarkRoast 对象、Mocha 对象和 Whip 对象，调用初始化函数，并输出。在 Mocha 类和 Whip 类中分别调用了基类的 getDescription()cost()。

【答案】(1)protected　　(2)virtual string getDescription　　(3)virtual int cost()=0

(4)Beverage* beverage　　(5)beverage　　(6)beverage

【典型题 10-34】(2013 年 5 月下午试题六)

阅读下列说明和 Java 代码，将应填入 (*n*) 处的字句写在答题纸的对应栏内。

【说明】

现要求实现一个能够自动生成求职简历的程序，简历的基本内容包括求职者的姓名、性别、年龄及工作经历。希望每份简历中的工作经历有所不同，并尽量减少程序中的重复代码。

现采用原型模式(Prototype)来实现上述要求，得到如图 10-37 所示的类图。

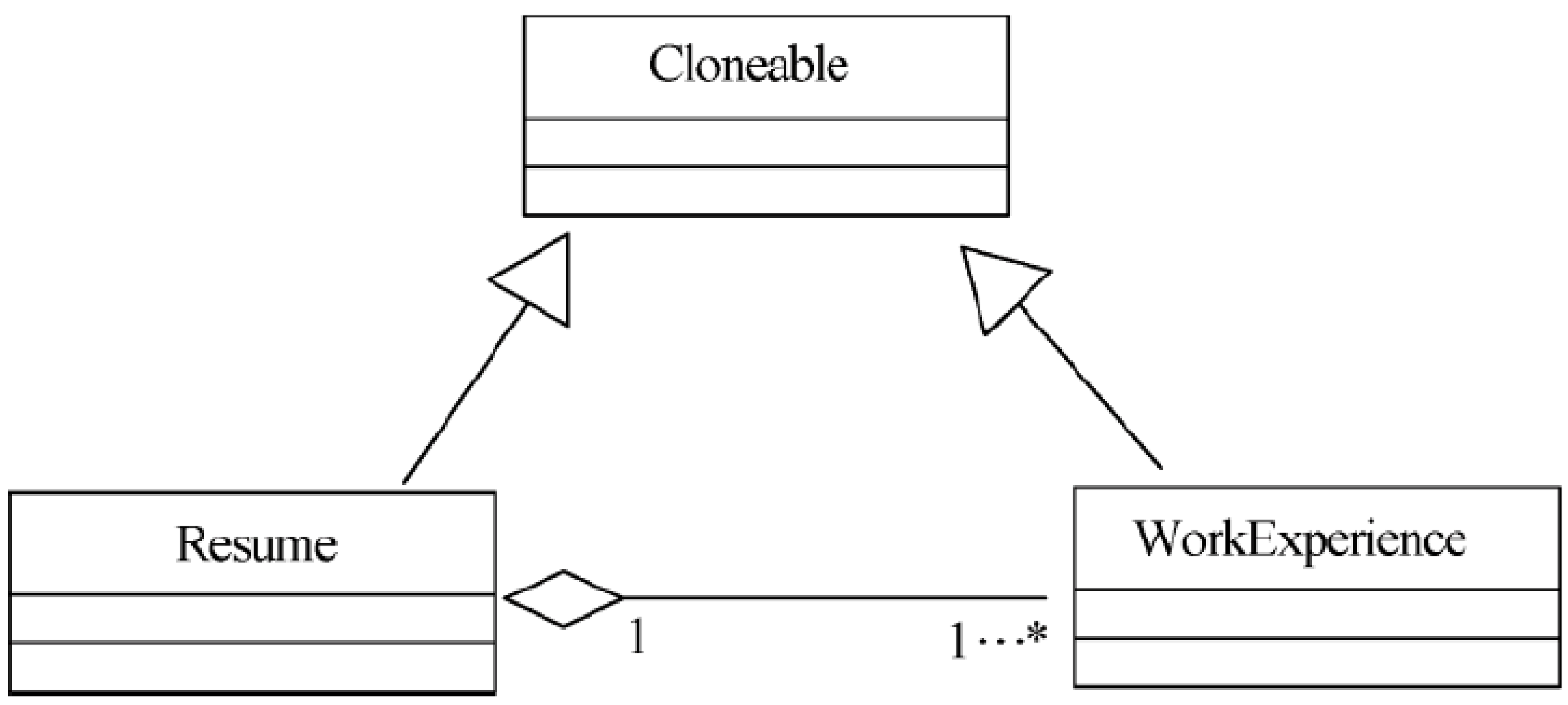


图 10-37　类图

【Java 代码】

```
class WorkExperience____(1)____Cloneable{  //工作简历
       private String workDate;
       private String company;
       public  Object  Clone() {
           ____________(2)________;
           obj.workDate=this.workDate;
           Obj.company-this.company;
           Return obj;
       }
}
class Resume ____(3)____  Cloneable{   //简历
      private String name;
      private String sex;
      private String age;
      private WorkExperience work;
      public Resume(String name){
        this.name=name;     work=new WorkExperience();
      }
      private Resume(WorkExperience work){
      this.woek= ____(4)____  ;
}
```

```
        public void SetPersonalInfo(String sex,String age){   /*代码省略*/ }
        public void SetWorkExperience(String workDate,String company){ /*代
码省略*/   }
        public Object Clone(){
            Resume obj= ____(5)____ ;
            //其余代码省略
            return  obj;
        }
}
class  WorkResume{
    public static void main(String[]  args){
        Resume  a=new  Resume(''张三'');
        a.SetPersonalInfo(''男'', ''29'');
        a.SetWorkExperience(''1998～2000'', ''XXX 公司'');
        Resume b=  ____(6)____ ;
        b.SetWorkExperience(''2001～2006'', ''YYY 公司'');
    }
}
```

【解析】

本题考查原型模式的概念及应用，原型模型是一种对象创建模型，用原型实体指定创建对象的种类，并且通过复制这些原型创建新的对象。原型模型允许一个对象再创建另一个可订制的对象，无须知道任何创建的细节。

所有的 Java 类都继承自 java.lang.Object，而 object 类提供一个 Clone()方法，可以将一个 Java 对象复制一份。因此在 Java 中可以直接使用 Object 提供的 Clone()方法来实现对象的克隆。能够实现克隆的 java 类必须实现一个标识接口 Cloneable，表示这个 java 类支持复制。

题中 WorkExperience 类和 Resume 类需要实现 Cloneable 接口，故(1)和(3)处为 implements，WorkExperience 中需要实现 Clone 方法，并将自身复制一份，由下面的代码可知(2)处为 WorkExperience obj=new WorkExperience()。Resume 类中的私有构造方法实现 WorkExperience 的深复制，故(4)处为(Work Experence)work.Clone( )，而 Resume 类中 Clone 方法实现自身的复制，故(5)处为 new Resume(this.work)

在 main 中实现 Resume b 对 a 的复制，故(6)处为 a.Clone( )。

【答案】

(1) implements	(2)WorkExperience obj=new WorkExperience()
(3) implements	(4)(WorkExperence)work.Clone( )
(5) new Resume(this.work)	(6)a.Clone()

【典型题 10-35】(2012 年 11 月下午试题六)

阅读下列说明 Java 代码，将应填入 __(n)__ 处的字句写在答题纸的对应栏内。

【说明】

现欲开发一个软件系统，要求能够同时支持多种不同的数据库，为此采用抽象工厂模式设计该系统。以 SQL Server 和 Access 两种数据库以及系统中的数据库表 Department 为例，其类图如图 10-38 所示。

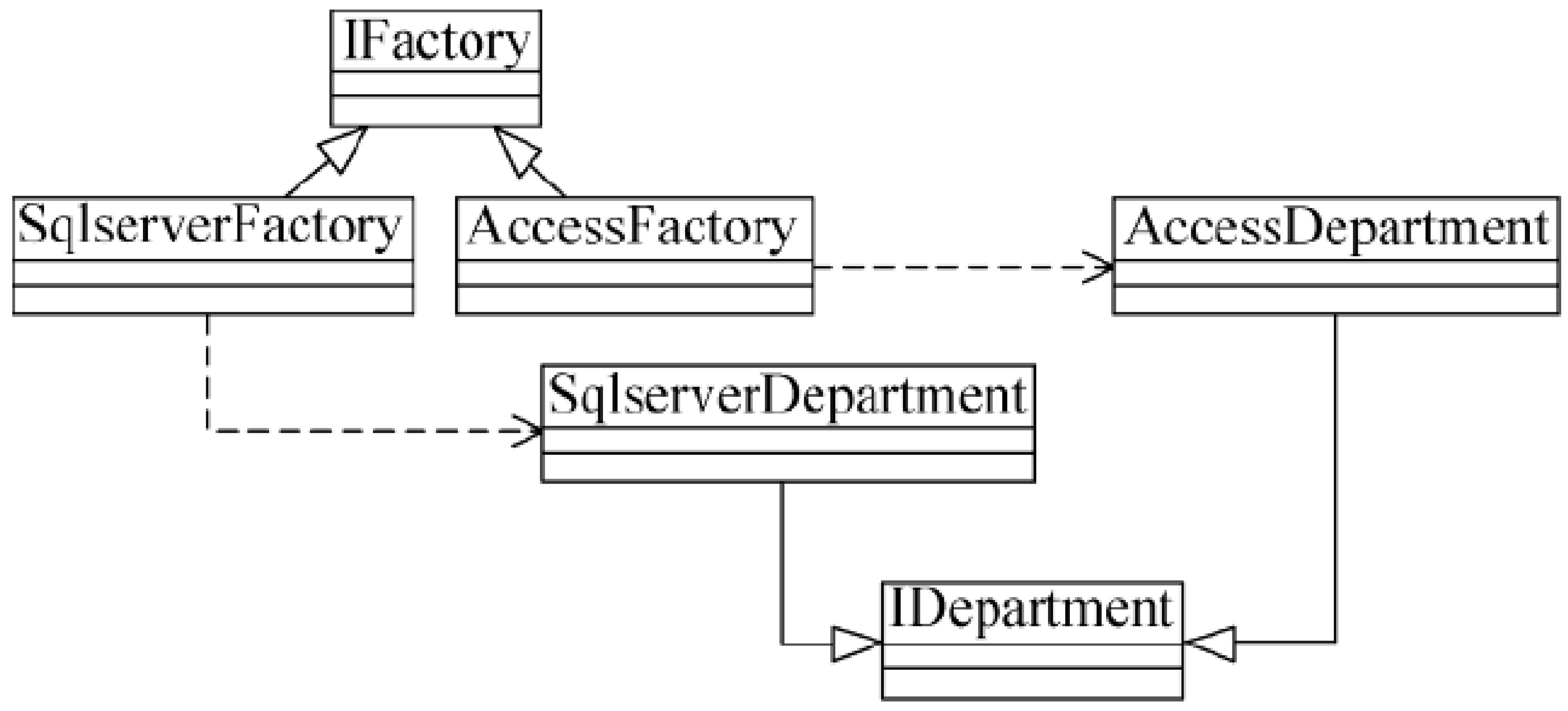


图 10-38　类图

【Java 代码】

```
import java.util.*;
class Department{/*代码省略*/}
interface IDepartment{
    ___(1)___;
    ___(2)___;
}
class SqlserverDepartment:___(3)___{
public:
    void Insert(Department department){
        System.out.println("Insert a record into Department in SQL
Server!\n");
        //其余代码省略
    }
    public Department GetDepartment(int id){
        /*代码省略*/
    }
}
class AccessDepartment:___(4)___{
    public void Insert(Department department){
        System.out.println("Insert a record into Department in ACCESS!\n");
        //其余代码省略
    }
    public Department GetDepartment(int id){
        /*代码省略*/
    }
};
___(5)___{
    ___(6)___;
}
class SqlServerFactory implements IFactory{
    public IDepartment CreateDepartment(){
        return new SqlserverDepartment();
    }
    //其余代码省略
};
class AccessFactory implements IFactory{
```

```
    public IDepartment CreateDepartment(){
        return new AccessDepartment();
    }
    //其余代码省略
};
```

【解析】

本题考查抽象工厂设计模式的概念及其应用。其类图如图 10-39 所示。

抽象工厂设计模式的意图是：提供一个创建一系列相关或相互依赖的对象，而无须指出它们具体的类。在如下情况下应当考虑使用抽象工厂模式：

- 当一个系统要独立于它的产品的创建、组合和表示时；
- 当一个系统要由多个产品系列中的一个来配置时；
- 当需强调一系列相关的产品对象的设计以便进行联合使用时；
- 想提供一组对象而不显示它们的实现过程，只显示它们的接口时。

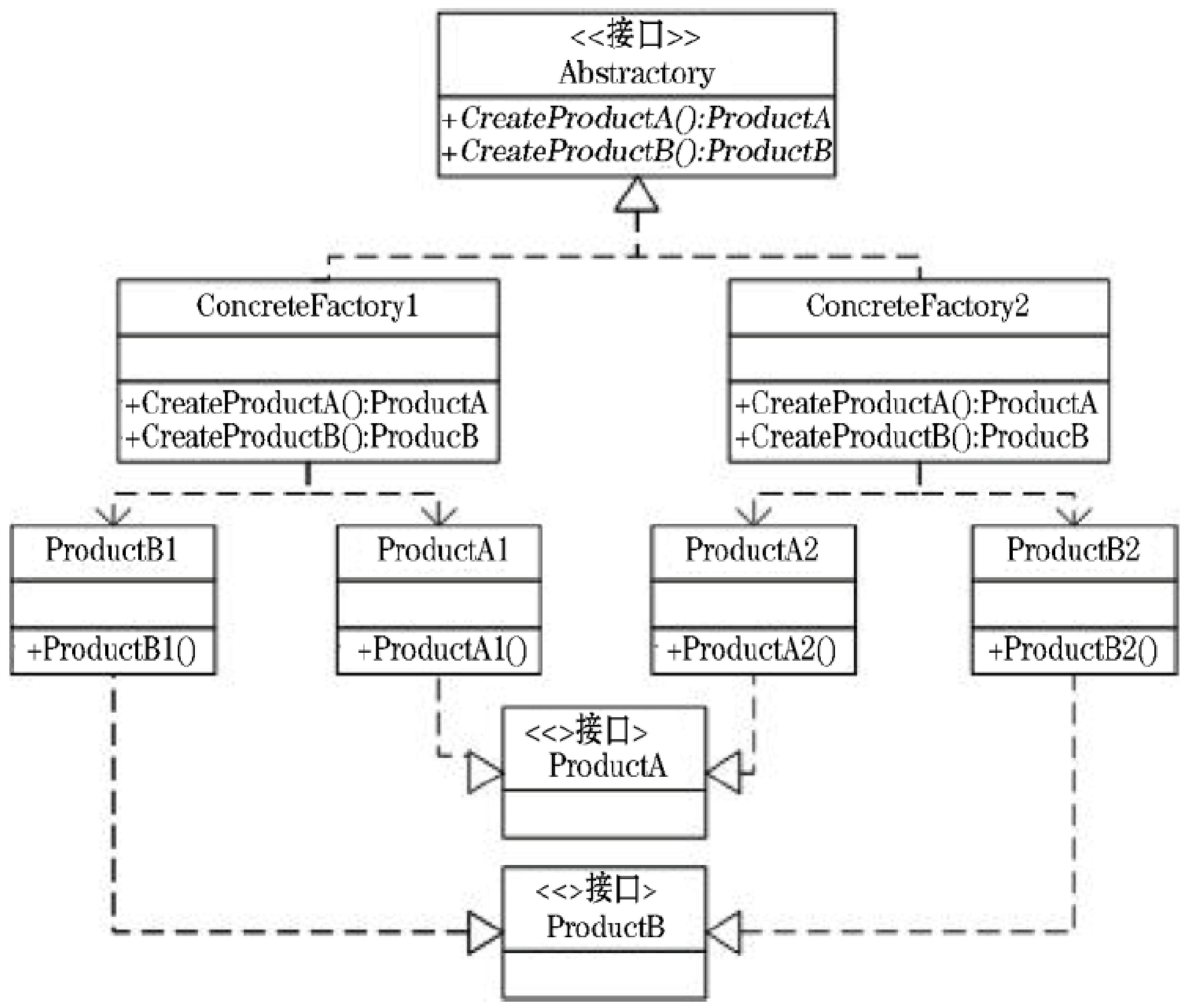


图 10-39　抽象工厂设计模式类图

抽象工厂设计模式的类图如下所示，其中：

- AbstractFactory 为抽象工厂，声明抽象产品的方法。
- ConcreteFactory 为具体工厂，执行生成抽象产品的方法，生成一个具体的产品。
- AbstractProduct 为抽象产品，为一种产品声明接口。
- Product 为具体产品，定义具体工厂生成的具体产品的对象，实现产品接口。

图 10-38 中的 IFactory 对应类图中的 AbstractFactory，Sqlserverfactory 和 Accessfactory 对应类图中的 ConcreteFactory，SqlserverDepartment 和 AccessDepartment 对应类图中的 Product，而 IDepartment 对应类图中的 AbstractProduct。

空(1)和空(2)考查接口 IDepartment 中方法的定义。由其子类 SqlserverDepartment 和 AccessDepartment

中方法的定义，可知空(1)处应填入 void Insert(Department department)，空(2)处应填入 Department GetDepartment(int id)。

空(3)和空(4)考查接口 IDepartment 的实现。接口的实现是在一个类的声明中使用关键字“implements”来表示该类使用某个已经定义的接口，然后即可在该类体中使用接口中定义的常量，而且必须实现接口中定义的所有方法。因此，空(3)和空(4)都应填入 implements IDepartment。

由于所给程序中缺少接口 IFactory 的定义，因此空(5)处应为定义接口 IFactory，应填入 interface IFactory。由于类 Sqlserverfactory 和 Accessfactory 中必须实现接口 IFactory 中定义的所有方法，观察这两个类中的方法可知，空(6)处应填入 IDepartment CreateDepartment()。

【答案】

(1) void Insert(Department department)

(2) Department GetDepartment(int id)

(3) implements IDepartment

(4) implements IDepartment

(5) interface IFactory

(6) IDepartment CreateDepartment()

【典型题 10-36】(2012 年 5 月下午试题六)

阅读下列说明和 Java 代码，将应填入__(*n*)__处的字句写在答题纸的对应栏内。

【说明】

某咖啡店在卖咖啡时，可以根据顾客的要求在其中加入各种配料，咖啡店会根据所加入的配料来计算费用。咖啡店所供应的咖啡及配料的种类和价格如表 10-10 所示。

表 10-10　咖啡及配料的种类和价格

咖　啡	价格/杯(￥)	配　料	价格/杯(￥)
蒸馏咖啡(Espresso)	25	摩卡(Mocha)	10
深度烘焙咖啡(DarkRoast)	20	奶泡(Whip)	8

现采用装饰器(Decorator)模式来实现计算费用的功能，得到如图 10-40 所示的类图。

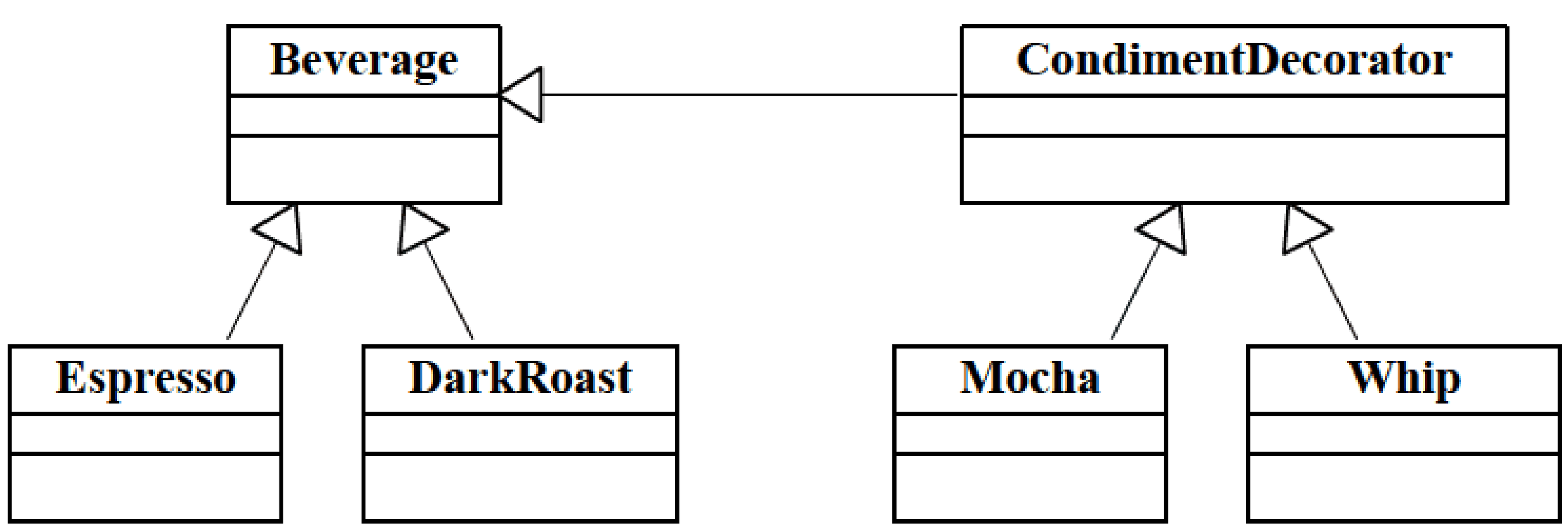


图 10-40　类图

【Java 代码】

```
import java.util.*;
  (1)   class Beverage { //饮料
    String description = "Unknown Beverage";
```

```
    public __(2)__ (){return description;}
    public __(3)__;
}
abstract class CondimentDecorator extends Beverage { //配料
    __(4)__ ;
}
class Espresso extends Beverage { //蒸馏咖啡
    private final int ESPRESSO_PRICE = 25;
    public Espresso() { description="Espresso"; }
    public int cost() { return ESPRESSO_PRICE; }
}
class DarkRoast extends Beverage { //深度烘焙咖啡
    private finalint DARKROAST_PRICE = 20;
    public DarkRoast() { description = "DarkRoast"; }
    public int cost(){ rcturn DARKROAST PRICE; }
}
class Mocha extends CondimentDecorator { //摩卡
    private final int MOCHA_PRICE = 10;
    public Mocha(Beverage beverage) {
        this.beverage = beverage;
    }
    public String getDescription() {
    return beverage.getDescription0 + ", Mocha";
    }
    public int cost() {
        return MOCHA_PRICE + beverage.cost();
    }
}
class Whip extends CondimentDecorator { //奶泡
    private finalint WHIP_PRICE = 8;
    public Whip(Beverage beverage) { this.beverage = beverage; }
    public String getDescription() {
        return beverage.getDescription()+", Whip";
    }
    public int cost() { return WHIP_PRICE + beverage.cost(); }
}
public class Coffee {
    public static void main(String args[]) {
        Beverage beverage = new DarkRoast();
        beverage=new Mocha( __(5)__ );
        beverage=new Whip ( __(6)__ );
        System.out.println(beverage.getDescription() +"￥" +beverage.cost());
    }
}
```

编译运行上述程序，其输出结果为：

```
DarkRoast, Mocha, Whip ￥38
```

【解析】装饰器模式描述了如何动态地为一个对象添加职责。该模式采用递归方式组合对象，从而允许添加任意多的对象职责。在下列情况下可以使用装饰器模式：

(1) 在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责。

(2) 处理那些可以撤销的职责。

(3) 当不能采用生成子类的方法进行扩充时。一种情况是，可能有大量独立的扩展，为支持每一种组合将产生大量的子类，使得子类数目呈爆炸性增长。另一种情况可能是因为类定义被隐藏，或类定义不能用于生成子类。

装饰器模式的结构如图 10-41 所示。其中：

- 抽象组件角色(Component)：定义一个对象接口，以规范准备接受附加责任的对象，即可以给这些对象动态地添加职责。
- 具体组件角色(ConcreteComponent)：被装饰者，定义一个将要被装饰增加功能的类。可以给这个类的对象添加一些职责。
- 抽象装饰器(Decorator)：维持一个指向构件 Component 对象的实例，并定义一个与抽象组件角色 Component 接口一致的接口。
- 具体装饰器角色(Concrete Decorator)：向组件添加职责。

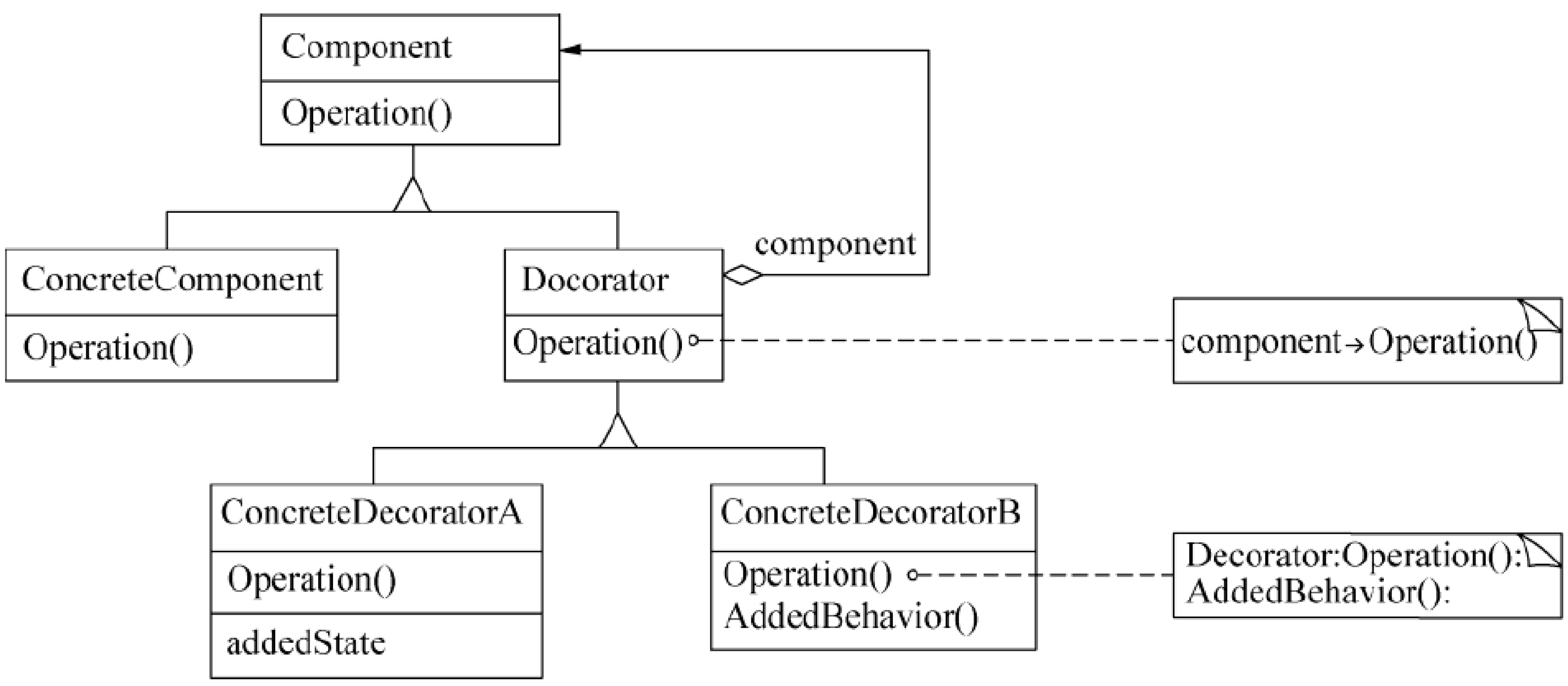


图 10-41　装饰器模式的结构

图 10-40 中的 Beverage 对应的就是图 10-41 中的抽象类 Component，Espresso 和 DarkRoast 对应的是 ConcreteComponent，CondimentDecorator 对应抽象装饰器 Decorator，Mocha 和 Whip 扮演的是类 CondimentDecorator 的具体装饰器角色 ConcreteDecorator。

由于类 Beverage 为其子类提供了统一的操作接口，所以将其定义为抽象类。通过可以在类名前加 abstract 关键字来定义抽象类，因此(1)处应填入 abstract。

(2)和(3)处考查构造函数的定义。从(2)处构造函数体中返回值的类型及后续的子类继承程序可知，(2)处应填入 String getDescription；从 public int cost() { return ESPRESSO_PRICE;}可以看出，cost()函数的返回值为常量，因此(3)处应填入 abstract int cost()。

(4)处考查对 CondimentDecorator 的定义，在该类中声明一类成员变量，并在 this.beverage = beverage 和 return beverage.getDescription0 + ", Mocha"中加以使用。因此(4)处应填入 Beverage beverage。

(5)和(6)处考查实例化类模板的方法。类模板必须在实例化后才能使用。实例化类模板时，要给出类型实参。从类图可知，(5)和(6)处均应填入 beverage。

【答案】(1)abstract　　(2)String getDescription　　(3)abstract int cost()

(4)Beverage beverage　(5)beverage　(6)beverage

10.10　过 关 习 题

1. 采用面向对象开发方法时，对象是系统运行的基本实体。以下关于对象的叙述中，正确的是______。

A. 对象只能包括数据(属性)　B. 对象只能包括操作(行为)

C. 对象一定有相同的属性和行为　D. 对象通常由对象名、属性和操作三个部分组成

2. 一个类是__(1)__。在定义类时，将属性声明为 private 的目的是__(2)__。

(1) A. 一组对象的封装　B. 表示一组对象的层次关系

C. 一组对象的实例　D. 一组对象的抽象定义

(2) A. 实现数据隐藏，以免意外更改　B. 操作符重载

C. 实现属性值不可更改　D. 实现属性值对类的所有对象共享

3. 在面向对象软件开发中，封装是一种______技术，其目的是使对象的使用者和生产者分离。

A. 接口管理　B. 信息隐藏　C. 多态　D. 聚合

4. 下列关于一个类的静态成员的描述中，不正确的是______。

A. 类的静态方法只能访问该类的静态数据成员

B. 静态数据成员可被该类的所有方法访问

C. 该类的对象共享其静态数据成员的值

D. 该类的静态数据成员的值不可修改

5. 开-闭原则(Open-Closed Principle，OCP)是面向对象的可复用设计的基石。开-闭原则是指一个软件实体应当对__(1)__开放，对__(2)__关闭；里氏代换原则(Liskov Substitution Principle，LSP)是指任何__(3)__可以出现的地方，__(4)__一定可以出现。依赖倒转原则(Dependence Inversion Principle，DIP)就是要依赖于__(5)__而不依赖于__(6)__，或者说要针对接口编程，不要针对实现编程。

(1) A. 修改　B. 扩展　C. 分析　D. 设计

(2) A. 修改　B. 扩展　C. 分析　D. 设计

(3) A. 变量　B. 常量　C. 基类对象　D. 子类对象

(4) A. 变量　B. 常量　C. 基类对象　D. 子类对象

(5) A. 程序设计语言　B. 建模语言　C. 实现　D. 抽象

(6) A. 程序设计语言　B. 建模语言　C. 实现　D. 抽象

6. 面向对象分析的第一步是______。

A. 定义服务　B. 确定附加的系统约束

C. 确定问题域　D. 定义类和对象

7. UML 中关联的多重度是指______。

A. 一个类中被另一个类调用的次数

B. 一个类的某个方法被另一个类调用的次数

C. 一个类的实例能与另一个类的多少个实例相关联

D. 两个类所具有的相同的方法和属性

8. __(1)__是一种很强的“拥有”关系，“部分”和“整体”的生命周期通常一样。整体对象完全支配

其组成部分，包括它们的创建和销毁等；__(2)__同样表示“拥有”关系，但有时候“部分”对象可以在不同的“整体”对象之间共享，并且“部分”对象的生命周期也可以与“整体”对象不同，甚至“部分”对象可以脱离“整体”对象而单独存在。上述两种关系都是__(3)__关系的特殊种类。

(1) A. 聚合　B. 组合　C. 继承　D. 关联

(2) A. 聚合　B. 组合　C. 继承　D. 关联

(3) A. 聚合　B. 组合　C. 继承　D. 关联

9. 采用 UML 进行面向对象开发时，部署图通常在______阶段使用。

A. 需求分析　B. 架构设计　C. 实现　D. 实施

10. 业务用例和参与者一起描述__(1)__，而业务对象模型描述__(2)__。

(1) A. 工作过程中的静态元素　B. 工作过程中的动态元素

C. 工作过程中的逻辑视图　D. 组织支持的业务过程

(2) A. 业务结构　B. 结构元素如何完成业务用例

C. 业务结构以及元素如何完成业务用例　D. 组织支持的业务过程

11. UML 的设计视图包含了类、接口和协作，其中，设计视图的静态方面由__(1)__和__(2)__表现；动态方面由交互图、__(3)__表现。

(1) A. 类图　B. 状态图　C. 活动图　D. 序列图

(2) A. 交互图　B. 对象图　C. 通信图　D. 定时图

(3) A. 状态图和类图　B. 类图和活动图

C. 对象图和状态图　D. 状态图和活动图

12. 在面向对象软件开发过程中，采用设计模式______。

A. 以复用成功的设计

B. 以保证程序的运行速度达到最优值

C. 以减少设计过程创建的类的个数

D. 允许在非面向对象程序设计语言中使用面向对象的概念

13. 以下关于 Singleton(单例)模式的描述中，正确的是______。

A. 它描述了只有一个方法的类的集合

B. 它描述了只有一个属性的类的集合

C. 它能够保证一个类的方法只能被一个唯一的类调用

D. 它能够保证一个类只产生唯一的一个实例

14. 欲动态地给一个对象添加职责，宜采用______模式。

A. 适配器(Adapter)　B. 桥接(Bridge)

C. 组合(Composite)　D. 装饰器(Decorator)

15. ______模式通过提供与对象相同的接口来控制对这个对象的访问。

A. 适配器(Adapter)　B. 代理(Proxy)

C. 组合(Composite)　D. 装饰器(Decorator)

16. 设计模式__(1)__将抽象部分与其实现部分相分离，使它们都可以独立地变化。图 10-42 是该设计模式的类图，其中__(2)__用于实现部分接口。

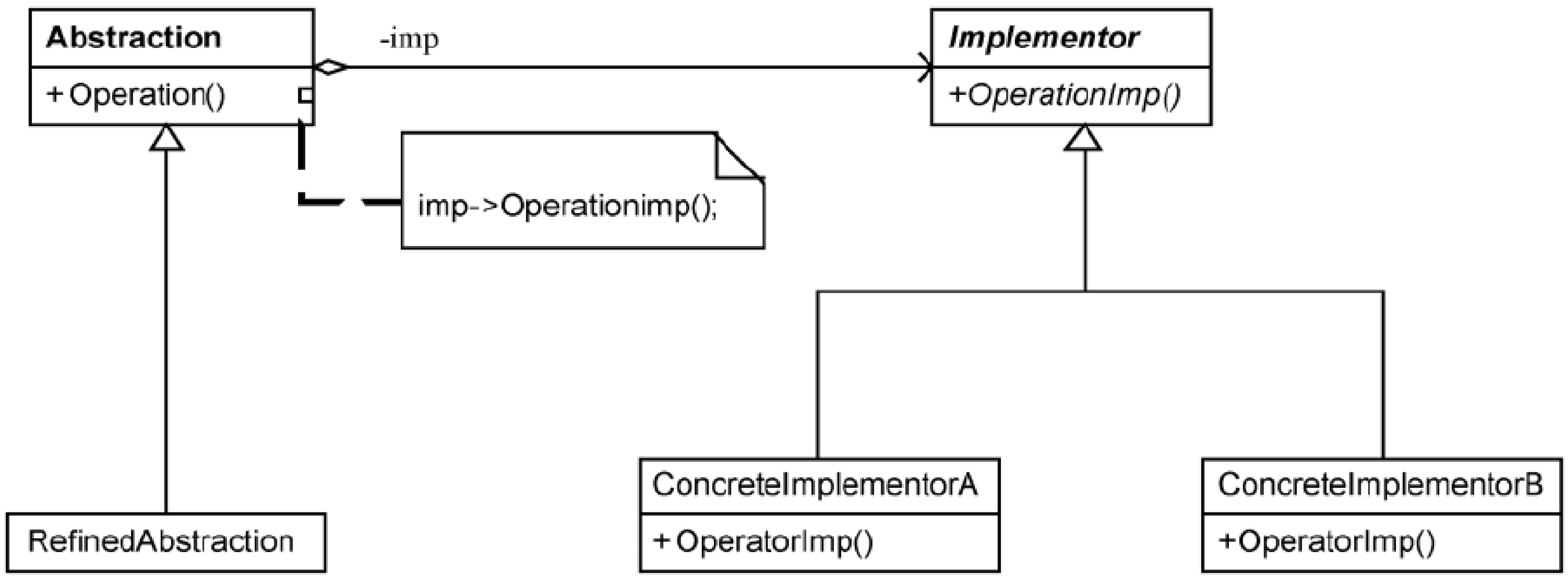


图 10-42　类图

(1)　A. Bridge(桥接)　　B. Composite(组合)　　C. Facade(外观)　　D. Singleton(单例)

(2)　A. Abstraction　　B. ConcreteImplementorA　　C. ConcreteImplementorB　　D. Implementor

17. ______将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

A. Adapter(适配器)模式　　B. Command(命令)模式

C. Singleton(单例)模式　　D. Strategy(策略)模式

18. __(1)__设计模式允许一个对象在其状态改变时，通知依赖它的所有对象。该设计模式的类图如图 10-43 所示，其中，__(2)__在其状态发生改变时，向它的各个观察者发出通知。

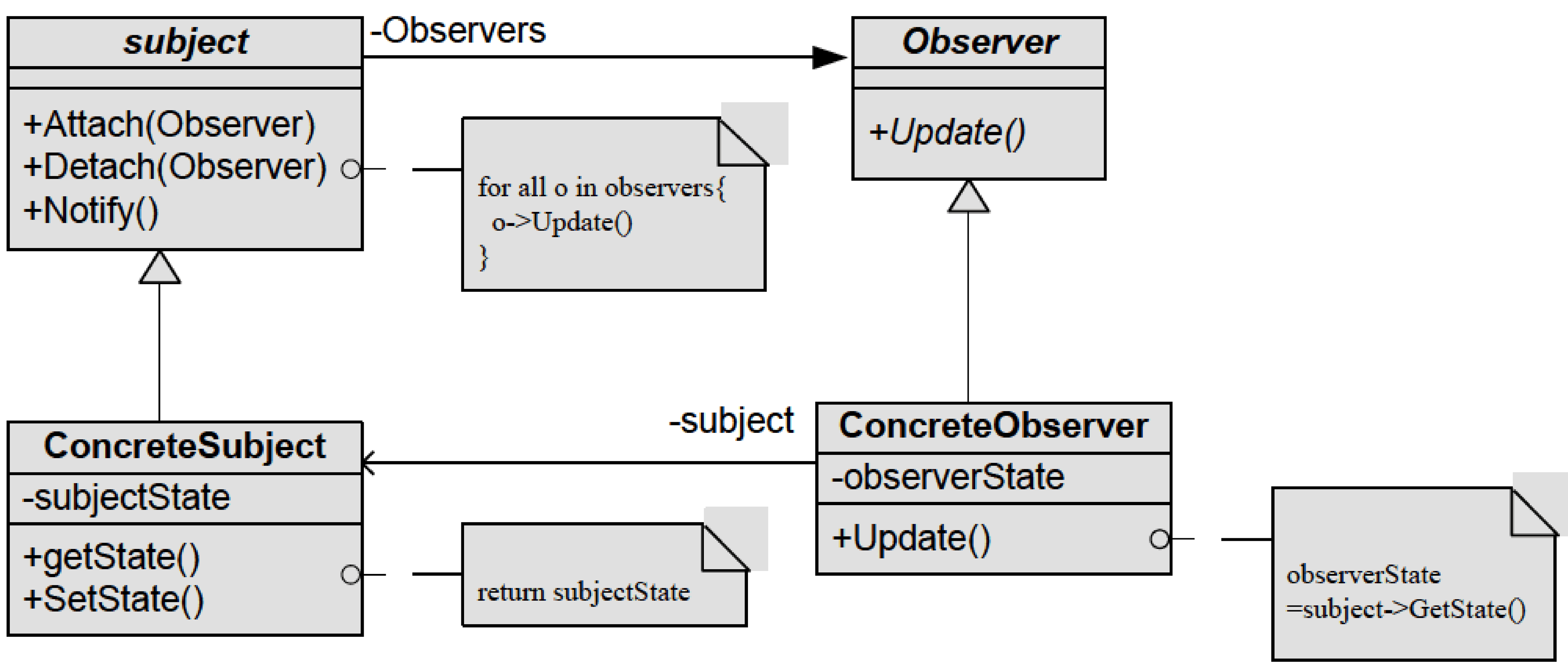


图 10-43　类图

(1)　A. 命令(Command)　　B. 责任链(Chain of Responsibility)

C. 观察者(Observer)　　D. 迭代器(Iterator)

(2)　A. Subject　　B. ConcreteSubject

C. Observer　　D. ConcreteObserver

19. 阅读下列说明和图，回答问题 1 至问题 3。

【说明】

Pay&Drive 系统(开多少付多少)能够根据驾驶里程自动计算应付的费用。

系统中存储了特定区域道路交通网的信息。道路交通网由若干个路段(Road Segment)构成，每个路段由两个地理位置坐标点(Node)标定，其里程数(Distance)是已知的。在某些地理坐标点上安装了访问控制(Access Control)设备，可以自动扫描行驶卡(Card)。行程(Trajectory)由一组连续的路段构成。行程的起点(Entry)和终点(Exit)都装有访问控制设备。

系统提供了三种行驶卡。常规卡(Regular Card)有效期(Valid Period)为一年，可以在整个道路交通网内使用。季卡(Season Card)有效期为三个月，可以在整个道路交通网内使用。单次卡(Minitrip Card)在指定的行程内使用，且只能使用一次。其中，季卡和单次卡都是预付卡(Prepaid Card)，需要客户(Customer)预存一定的费用。

系统的主要功能有客户注册、申请行驶卡、使用行驶卡行驶等。

使用常规卡行驶，在进入行程起点时，系统记录行程起点、进入时间(Date of Entry)等信息。在到达行程终点时，系统根据行驶的里程数和所持卡的里程单价(Unit Price)计算应付费用，并打印费用单(Invoice)。

季卡的使用流程与常规卡类似，但是不需要打印费用单，系统自动从卡中扣除应付费用。

单次卡的使用流程与季卡类似，但还需要在行程的起点和终点检查行驶路线是否符合该卡所规定的行驶路线。

现采用面向对象方法开发该系统，使用 UML 进行建模。构建出的用例图和类图分别如图 10-44 和图 10-45 所示。

【问题 1】

根据说明中的描述，给出图 10-44 中的 U1 和 U2 所对应的用例，以及(1)所对应的关系。

【问题 2】

根据说明中的描述，给出图 10-45 中缺少的 C1～C6 所对应的类名以及(2)～(3)处所对应的多重度(类名使用说明中给出的英文词汇)。

【问题 3】

根据说明中的描述，给出 RoadSegment、Trajectory 和 Card 所对应的类的关键属性(属性名使用说明中给出的英文词汇)。

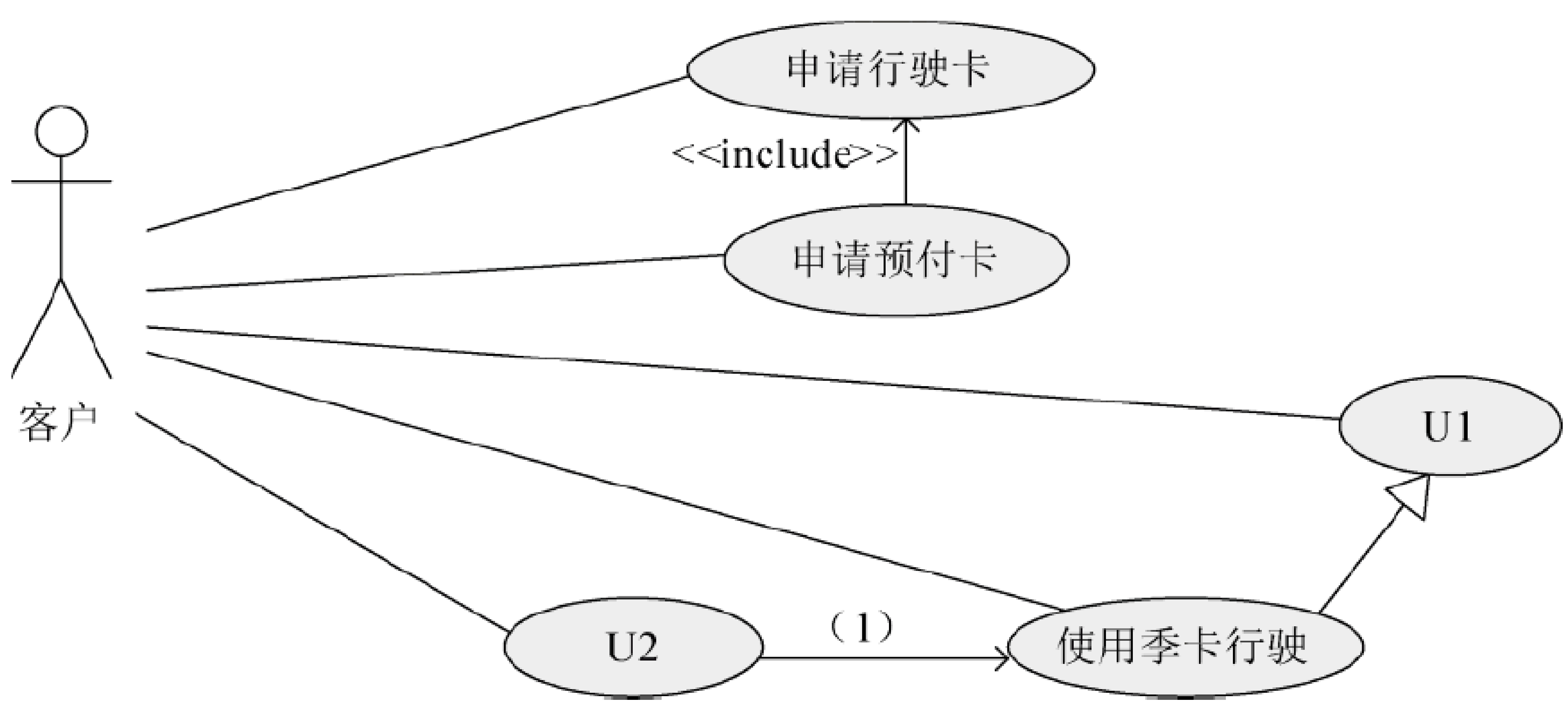


图 10-44　用例图

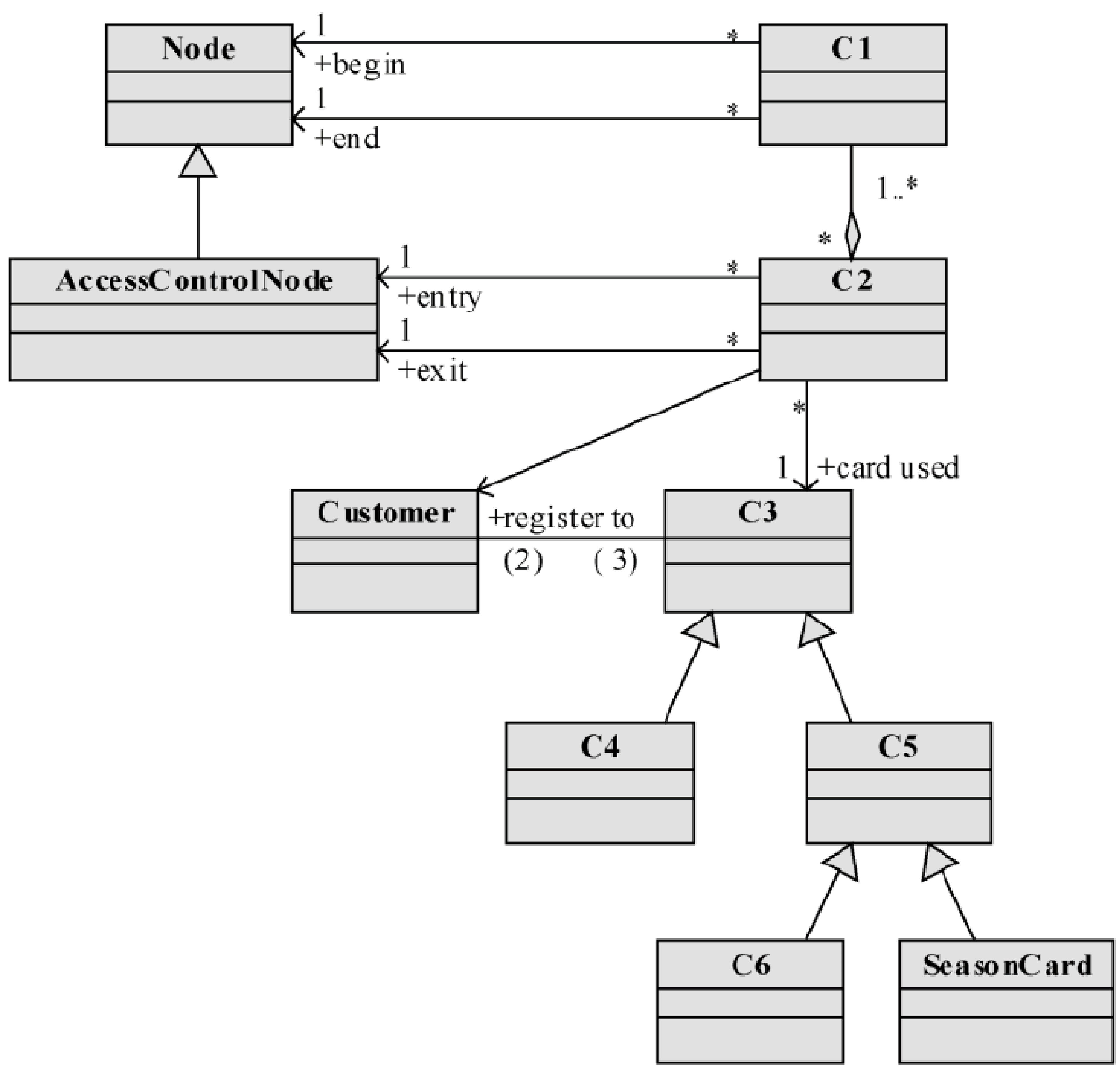


图 10-45　类图

20. 阅读下列说明和图，回答问题 1 至问题 3。

【说明】

一个简单的图形编辑器提供给用户的基本操作包括创建图形、创建元素、选择元素以及删除图形。图形编辑器的组成及其基本功能描述如下。

(1) 图形由文本元素和图元元素构成，图元元素包括线条、矩形和椭圆。

(2) 图形显示在工作空间中，一次只能显示一张图形(即当前图形，Current)。

(3) 编辑器提供了两种操作图形的工具：选择工具和创建工具。对图形进行操作时，一次只能使用一种工具(即当前活动工具，Active)。

① 创建工具用于创建文本元素和图形元素。

② 对于显示在工作空间中的图形，使用选择工具能够选定其中所包含的元素，可以选择一个元素，也可以同时选择多个元素。被选择的元素成为当前选中元素(Selected)。

③ 每种元素都具有相应的控制点。拖拽选定元素的控制点，可以移动元素或者调整元素的大小。

现采用面向对象方法开发该图形编辑器，使用 UML 进行建模。构建出的用例图和类图分别如图 10-46 和图 10-47 所示。

【问题 1】

根据说明中的描述，给出图 10-46 中 U1 和 U2 所对应的用例，以及(1)和(2)处所对应的关系。

【问题 2】

根据说明中的描述，给出图 10-47 中缺少的 C1～C8 所对应的类名以及(3)～(6)处所对应的多重度。

【问题 3】

图 10-47 中的类图设计采用了桥接(Bridge)设计模式，说明该模式的内涵。

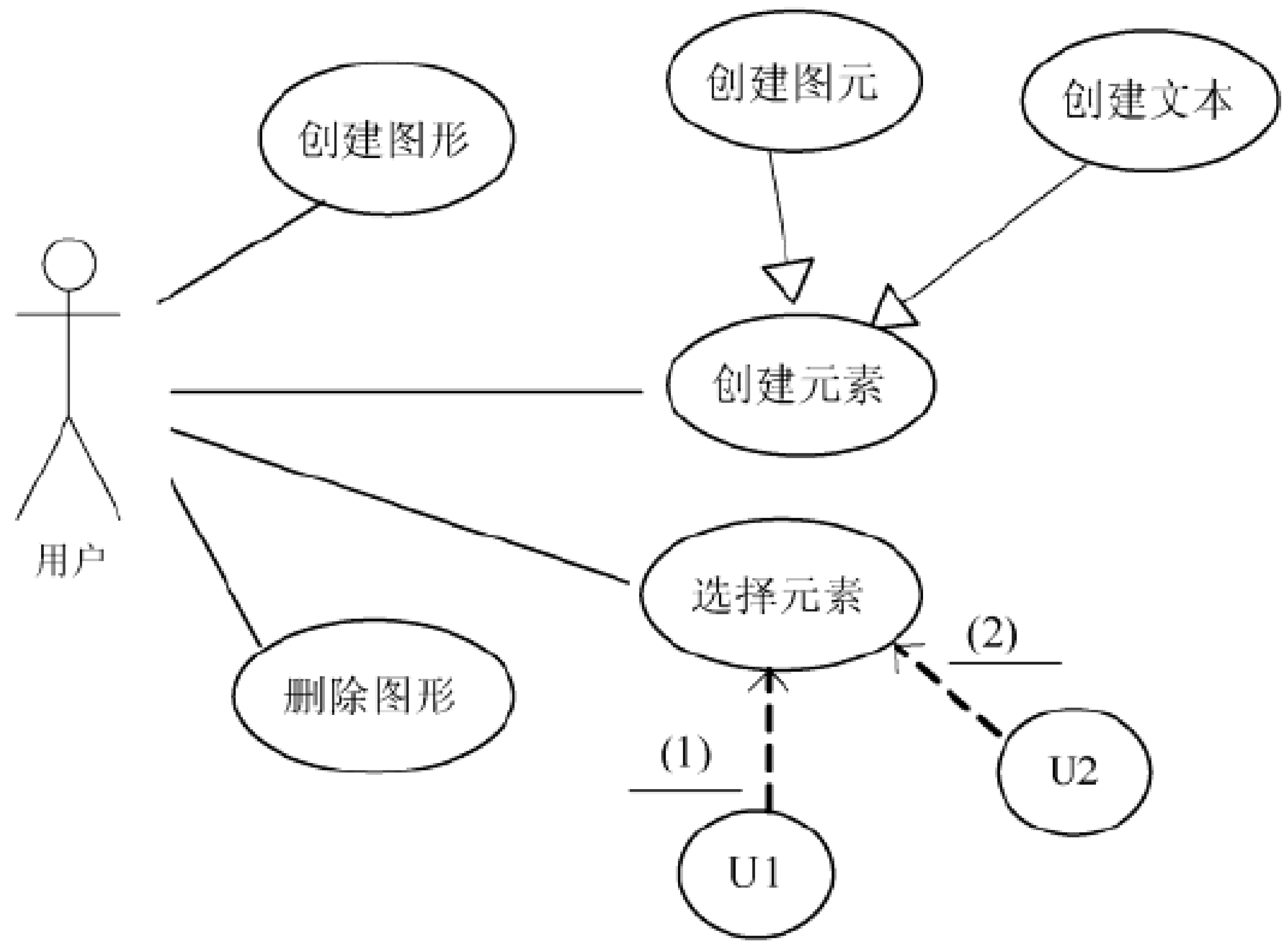


图 10-46　用例图

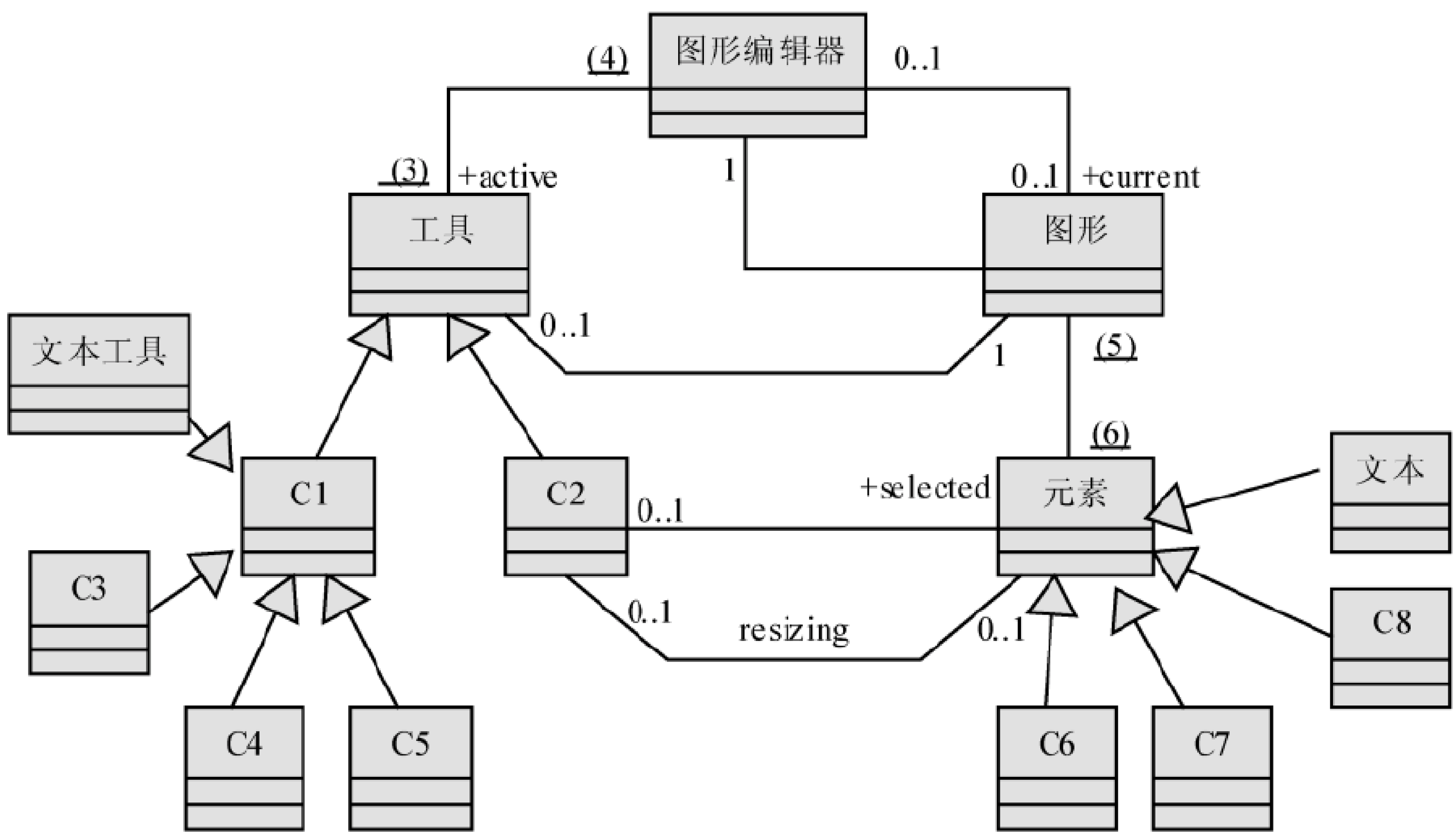


图 10-47　类图

21．阅读下列说明和 C++代码，将应填入 __(n)__ 处的字句写在答题纸的对应栏内。

【说明】

某饭店在不同的时段提供多种不同的餐饮，其菜单的结构如图 10-48 所示。

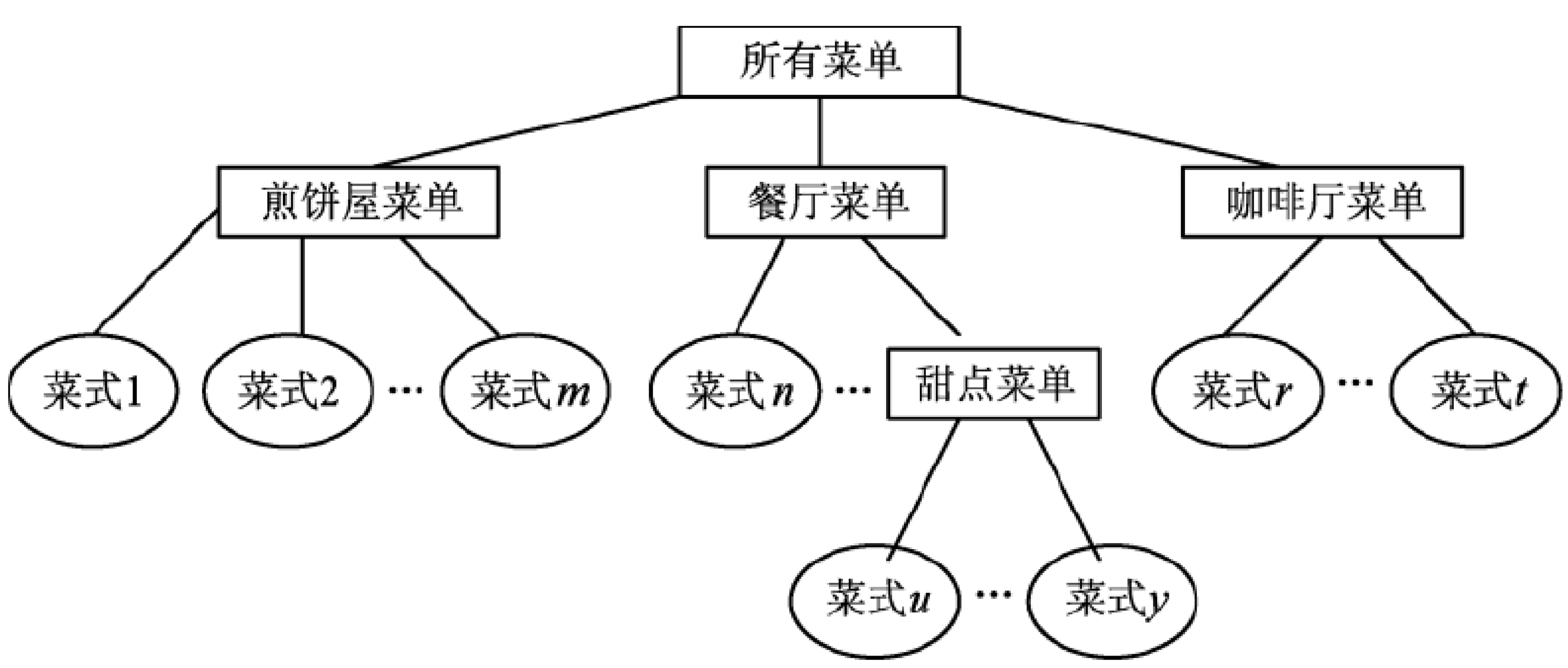


图 10-48　菜单结构

现在采用组合(Composition)模式来构造该饭店的菜单，使得饭店可以方便地在其中添加新的餐饮形式，得到如图 10-49 所示的类图。其中 MenuComponent 为抽象类，定义了添加(Add)新菜单和打印饭店所有菜单信息(Print)的方法接口。类 Menu 表示饭店提供的每种餐饮形式的菜单，如煎饼屋菜单、咖啡屋菜单等。每种菜单中都可以添加子菜单，例如图 10-49 中的甜点菜单，类 MenuItem 表示菜单中的菜式。

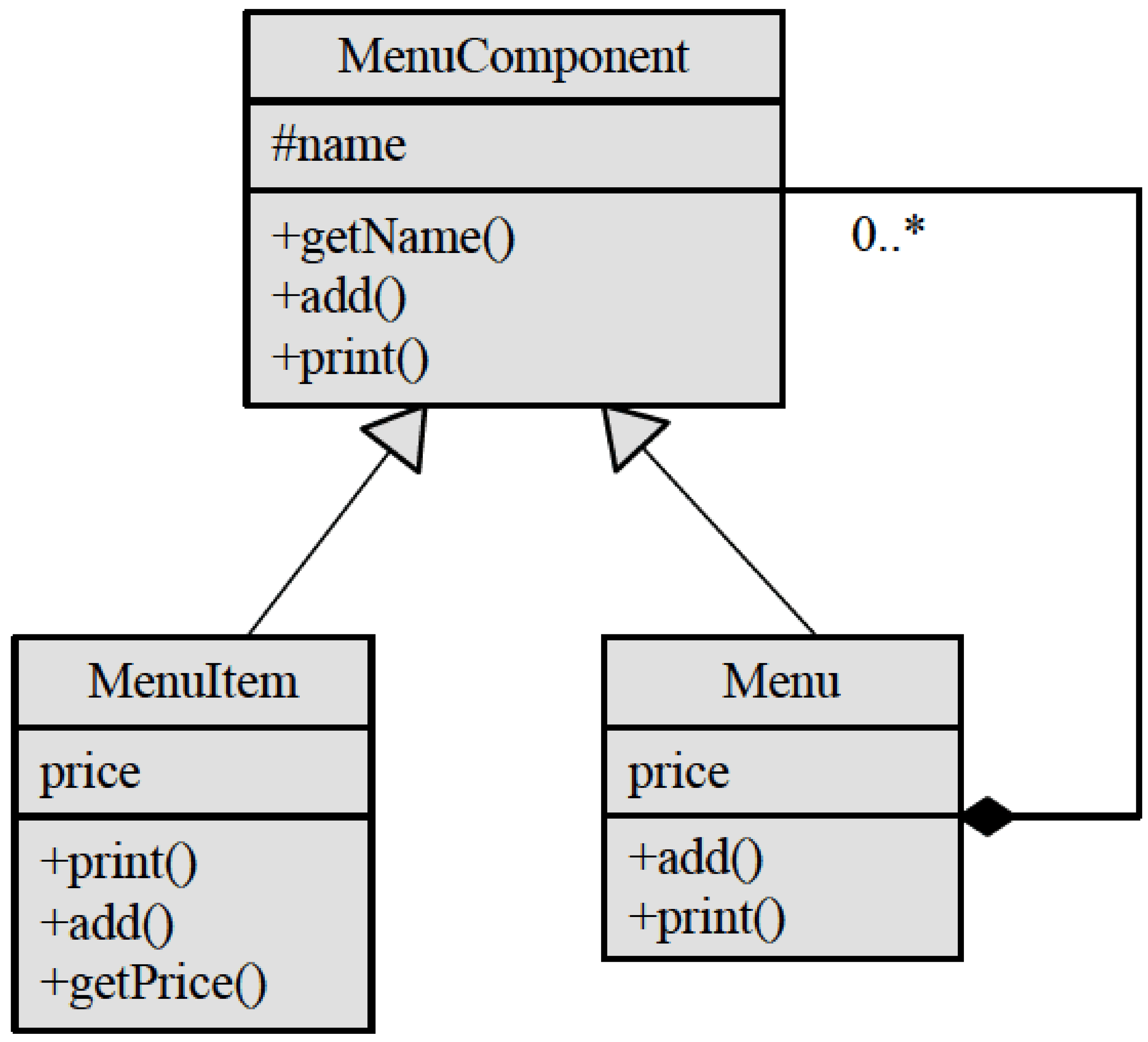


图 10-49　类图

【C++代码】

```
#include<iostream>
#include<list>
#include<string>
using namespace std;
class MenuComponent{
protected: string name;
public:
    MenuComponent(string name){ this->name=name; }
    sting getName(){ return name; }
    ___(1)___;  //添加新菜单
    virtual void print()=0;   //打印菜单信息
};
class MenuItem:public MenuComponent{
private:double price;
public:
    MenuItem(string name, double price):MenuComponent(name){ this->price=
    price;}
    double getPrice(){return price;}
    void add(MenuComponent* menuComponent){return;}  //添加新菜单
    void print(){cout<<"  "<<getName()<<","<<getPrice<<endl;}
};
class Menu:public MenuComponent{
private: list<___(2)___>menuComponents;
public:
    Menu(string name):MenuComponent(name){}
```

```
        void add(MenuComponent* menuComponent)  //添加新菜单
        { __(3)__; }
        void print(){
            cout<<"\n"<<getName()<<"\n----------------"<<endl;
            std::list<MenuComponent *>::iterator iter;
            for(iter=menuComponents.begin();iter!=menuComponents.end(); iter++)
               __(4)__ ->print();
        }
    };
    void main(){
        MenuComponent* allMenus=new Menu("ALL MENUS");
        MenuComponent* dinerMenu=new Menu("DINER MENU");
        … //创建更多的 Menu 对象，此处代码省略
        allMenus->add(dinerMenu);  //将 dinerMenu 添加到餐厅菜单中
        … //为餐厅增加更多的菜单，此处代码省略
      __(5)__ ->print(); //打印饭店所有菜单信息
    }
```

22. 阅读下列说明和 Java 代码，将应填入(*n*)处的字句写在答题纸的对应栏内。

【说明】

某大型商场内安装了多个简易的纸巾售卖机，自动售出 2 元钱一包的纸巾，且每次仅售出一包纸巾。纸巾售卖机的状态如图 10-50 所示。

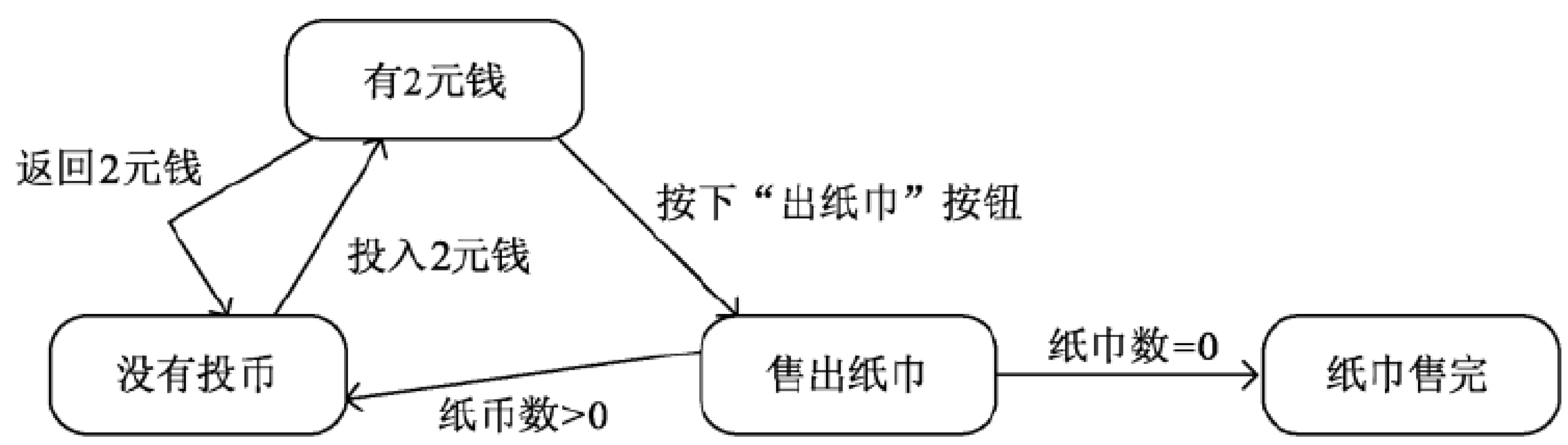


图 10-50　纸巾售卖机状态

采用状态(State)模式来实现该纸巾售卖机，得到如图 10-51 所示的类图。其中类 State 为抽象类，定义了投币、退币、出纸巾等方法接口。类 SoldState、SoldOutState、NoQuarterState 和 HasQuarterState 分别对应图 10-50 中纸巾售卖机的四种状态：售出纸巾、纸巾售完、没有投币、有 2 元钱。

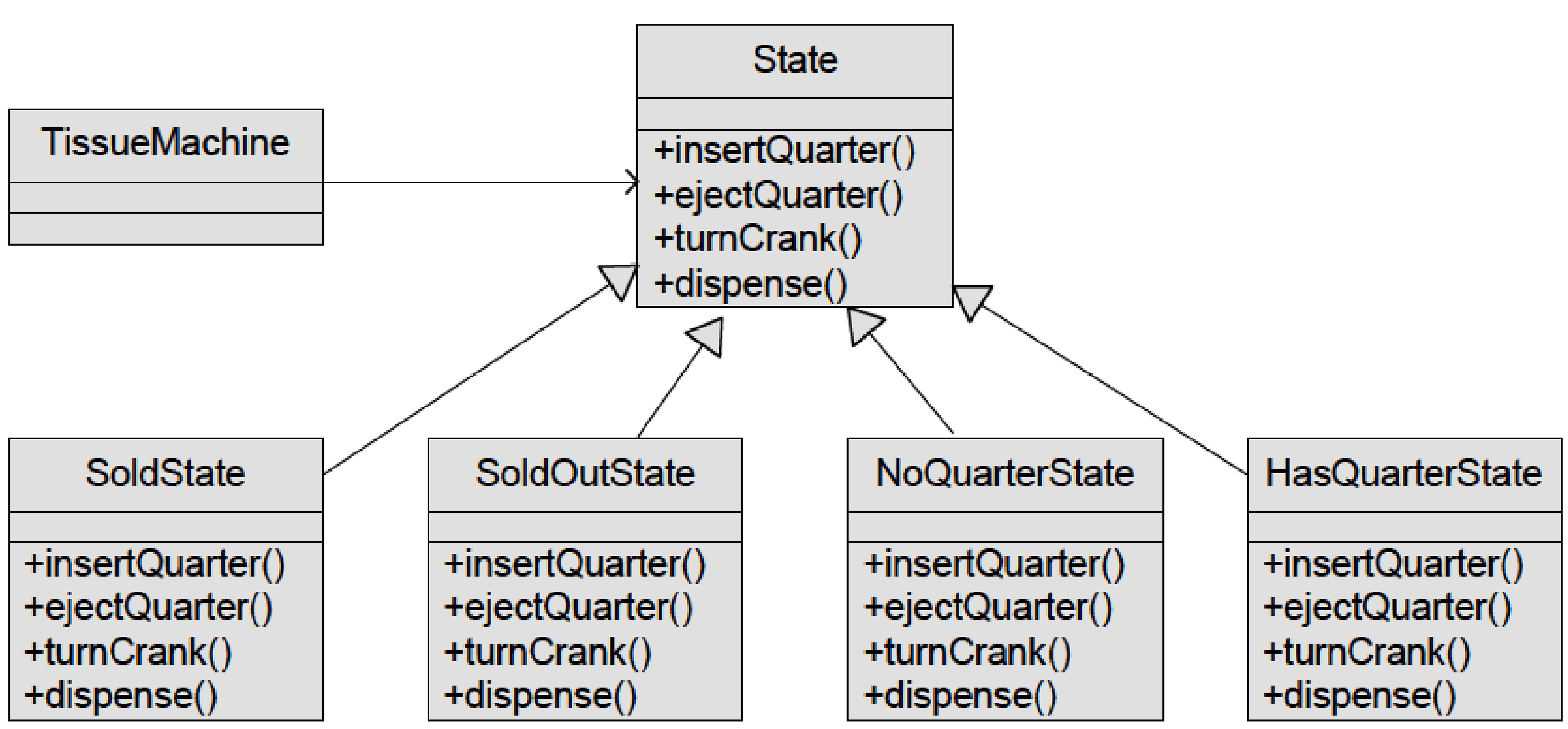


图 10-51　类图

【Java 代码】

```
import java.util.*
interface State{
    public void insertQuarter();   //投币
    public void ejectQuarter();    //退币
    public void turnCrank();       //按下“出纸巾”按钮
    public void dispense();        //出纸巾
}
class TissueMachine{
    __(1)__ soldOutState,noQuarterState,hasQuarterState,soldState,state;
    state=soldOutState;
    int count=0;   //纸巾数
    public TissueMachine(int number) {/*实现代码省略*/}
    public State getHasQuarterState() {return hasQuarterState;}
    public State getNoQuarterState() {return noQuarterState;}
    public State getSoldState() {return soldState;}
    public State getSoldOutState() {return soldOutState;}
    int getCount {return count;}
    //其余代码省略
};
class NoQuarterState implement State {
   TissueMachine tissueMachine;
   public void insertQuarter(){
       tissureMachine.setState(__(2)__);
   }
//构造方法以及其余代码省略
}
class HasQuarterState implement State {
   TissueMachine tissueMachine;
   public void ejectQuarter(){
       tissureMachine.setState(__(3)__);
   }
//构造方法以及其余代码省略
}
class SoldState implement State {
   TissueMachine tissueMachine;
   public void dispense(){
       if(tissueMachine.getCount()>0){
          tissureMachine.setState(__(4)__);
       }
       else{
          tissureMachine.setState(__(5)__);
       }
   }
}
```

第 11 章

标准化和软件知识产权基础

本章要点

- 保护知识产权有关的法律、法规
- 计算机软件著作权
- 商业秘密权
- 专利权

11.1 标准化基础知识

考核说明：本节主要介绍标准化的基本概念、信息技术标准化、标准化组织等内容，在最近几次考试中没有出现相关考点，了解即可。

11.1.1 标准化的基本概念

1. 标准、标准化的概念

标准是对重复性事物或概念所做的统一规定。其主要形式有规范和规程。

标准化是在经济、技术、科学及管理等社会实践中，以改进产品、过程和服务的适用性，防止贸易壁垒，促进技术合作，促进最大社会效益为目的，对重复性事物和概念通过制定、发布和实施标准达到统一，获得最佳秩序和社会效益的过程。

2. 标准化的范围和对象

标准化的范围包括生产、经济、技术、科学及管理等社会实践中具有重复性的事物和概念以及需要建立统一技术要求的各个领域。

标准化对象分为两大类，一类是标准化的具体对象，即需要制定标准的具体事物；另一类是标准化的总体对象，即各种具体对象的全体所构成的整体。

3. 标准化过程模式

标准是标准化活动的产物，其目的和作用都是通过制定和贯彻具体的标准来体现的。标准化不是一个孤立的事物，而是一个活动过程。标准化活动过程一般包括标准产生(调查、研究、形成草案、批准发布)子过程、标准实施(宣传、普及、监督、咨询)子过程和标准更新(复审、废止或修订)子过程等。

4. 标准的分类

我国标准分为国家标准、行业标准、地方标准和企业标准四级。对需要在全国范畴内统一的技术要求，应当制定国家标准。对没有国家标准而又需要在全国某个行业范围内统一的技术要求，可以制定行业标准。对没有国家标准和行业标准而又需要在省、自治区、直辖市范围内统一的工业产品的安全、卫生要求，可以制定地方标准。

5. 标准的代号和编号

1) ISO 的代号和编号

ISO 的代号和编号的格式：ISO+标准号+[杠+分类号] +冒号+发布年号(方括号内的内容可有可无)。

2) 国家标准的代号和编号

强制性国家标准代号为 GB，推荐性国家标准的代号为 GB/T。

国家标准的编号由国家标准的代号、标准发布顺序号和标准发布年代号组成。

- 强制性国家标准：GB ×××××—×××××。
- 推荐性国家标准：GB/T ×××××—×××××。

3) 行业标准的代号和编号

行业标准代号由国家主管部门审查批准公布，已公布的有 QJ(航天)、SJ(电子)、JB(机械)、JR(金融系统)等。

行业标准编号由行业标准代号、标准发布顺序及标准发布年代号组成。

- 强制性行业标准：×× ×××××—×××××。
- 推荐性行业标准：××/T ×××××—×××××。

4) 地方标准的代号和编号

地方标准的代号：由大写字母 DB 加上省、自治区、直辖市行政区划代码的前两位数字，再加上“/T”组成推荐性地方标准，不加“/T”的为强制性标准。

地方标准的编号：由地方标准代号、地方标准发布顺序号、标准发布年代号组成，表示方法如下。

- 强制性地方标准：DB×× ×××—××××。
- 推荐性地方标准：DB××/T ×××—××××。

5) 企业标准的代号和编号

企业标准的代号由大写字母 Q 加斜线再加企业代号组成。

企业标准的编号由企业标准代号、标准发布顺序号和标准发布年代号组成，表示方法为“Q/××× ××××—××××”。

6. 信息技术标准化

- 信息编码标准化。
- 汉字编码标准化。
- 软件工程标准化。

11.1.2 标准化组织

1. 国际标准化组织

- 国际标准化组织(ISO)。
- 国际电工委员会(IEC)。

2. 区域标准化组织

- 欧洲标准化委员会(CEN)。
- 欧洲电工标准化委员会(CENELEC)。
- 亚洲标准咨询委员会(ASAC)。
- 国际电信联盟 (ITU)。

3. 行业标准化组织

(1) 美国电气电子工程师学会(IEEE)，IEEE 公布的标准常冠有 ANSI 字头。

(2) 美国国防部批准和颁布适用于美国军队内部使用的标准，代号为 DOD(采用公制计量单位的以 DOD 表示)和 MIL。

(3) 我国国防科学技术委员会批准和颁布适用于国防部门和军队使用的标准，代号为 GJB。例如，1988 年发布实施的 GJB 437—88 军用软件开发规范。

4. 国家标准化组织

- 美国国家标准学会(ANSI)。
- 英国标准学会(BSI)。
- 德国标准化学会(DIN)。
- 法国标准化协会(AFNOR)。

11.1.3 ISO9000 标准简介

1. ISO9000 标准

ISO9000 标准是一系列标准的统称，是由 ISO/TC176 制定的所有国际标准。TC176 专门负责制定品质管理和品质保证技术的标准。2000 年 12 月 15 日，ISO9000: 2000 系列标准正式发布实施。ISO9000:2000 系列标准采用了以过程为基础的质量管理体系机构模式，在标准构思和标准目的等方面体现了具有时代气息的变化，还将持续改进的思想贯穿于整个标准，把组织的质量管理体系满足顾客要求的能力和程度体现在标准的要求之中。

2. ISO9000: 2000 系列标准文件结构

ISO9000: 2000 系列标准现有 13 项标准，由 4 个核心标准、1 个支持标准、6 个技术报告、3 个小册子和 1 个技术规范构成，ISO9000、ISO9001、ISO9004 和 ISO19011 四项标准是 ISO9000 族标准的核心标准。

3. ISO9000: 2000 核心标准

ISO9000:2000 的核心标准如下。

- ISO9000: 2000《质量管理体系——基础和术语》。
- ISO9001: 2000《质量管理体系——要求》。
- ISO9004: 2000《质量管理体系——业绩改进指南》。
- ISO19011: 2000《质量管理体系和环境管理体系审核指南》。

11.1.4 ISO/IEC15504 过程评估标准简介

在 ISO/IEC15504 中关于过程评估的文档主要有如下几种。

- 概念和绪论指南。
- 过程和过程能力参考模型。
- 实施评估。
- 评估实施指南。

- 评估模型和标志指南。
- 评估师能力指南。
- 过程改进应用指南。
- 确定供方能力应用指南。
- 词汇。

11.2　知识产权基础知识

考核说明： 本节主要介绍知识产权的基本概念、计算机软件著作权、商业秘密权、专利权和商标权等，是考核的重点，每次考试都有 2～3 道题目，特别需要掌握各种权利的形成条件和归属。

11.2.1　知识产权的基本概念

1. 工业产权的概念

知识产权又称为智慧财产权，是指人们通过自己的智力活动创造的成果和经营管理活动中的经验、知识而依法所享有的权利。传统的知识产权可分为“工业产权”和“著作权”(版权)两类。世界贸易组织(WTO)的与贸易有关的知识产权协议(TRIPS)还把“未披露过的信息专有权”(商业秘密)、“集成电路布图设计权”列为知识产权的范围。

我国承认并以法律形式加以保护的主要知识产权为著作权、专利权、商标权、商业秘密、其他有关知识产权。

1)　工业产权

工业产权包括专利、实用新型、工业品外观设计、商标、服务标记、厂商名称、产地标记或原产地名称、制止不正当竞争等项内容。此外，商业秘密、微生物技术、遗传基因技术等也属于工业产权保护的对象。

2)　著作权

著作权(又称为版权)是指作者对其创作的作品享有的人身权和财产权，包括发表权、署名权、修改权和保护作品完整权、复制权、发行权、出租权、展览权、表演权、放映权、广播权、信息网络传播权、摄制权、改编权、翻译权、汇编权、应当由著作权人享有的其他权利。著作权的保护对象包括文学、科学和艺术领域内的一切作品。

有些智力成果可以同时成为这两类知识产权保护的客体。例如，计算机软件和实用艺术品属著作权保护的同时，权利人还可以申请发明专利或外观设计专利，获得专利权，成为工业产权保护的对象。

2. 知识产权的特点

知识产权具有以下特点。

- 无形性。
- 双重性。

- 确认性。
- 独占性。
- 地域性。
- 时间性。

3. 我国保护知识产权的法规

我国保护知识产权方面主要有如下法规。

- 《中华人民共和国著作权法》
- 《中华人民共和国专利法》
- 《中华人民共和国继承法》
- 《中华人民共和国公司法》
- 《中华人民共和国合同法》
- 《中华人民共和国产品质量法》
- 《中华人民共和国反不正当竞争法》
- 《中华人民共和国刑法》
- 《中华人民共和国计算机信息系统安全保护条例》
- 《中华人民共和国计算机软件保护条例》
- 《中华人民共和国著作权法实施条例》

真题链接

【例 11-1】王某买了一幅美术作品原件，则他享有该美术作品的<u>(10)</u>。(2014 年 5 月真题 10)

(10) A. 著作权 B. 所有权 C. 展览权 D. 所有权与其展览权

【解析】很显然，作品的买卖导致了所有权的转移，著作权法第十八条规定：“美术等作品原件所有权的转移，不视为作品著作权的转移，但美术作品原件的展览权由原件所有人享有。”因此，作品交易后，著作权仍归原作者，王某享有购买的美术作品的所有权和其展览权。

【答案】D

11.2.2 计算机软件著作权

1. 计算机软件著作权的主体与客体

1) 计算机软件著作权的主体

计算机软件著作权的主体指享有著作权的人，包括公民、法人和其他组织。

2) 计算机软件著作权的客体

计算机软件著作权的客体指著作权法保护的计算机软件著作权的范围，根据《著作权法》第三条和《计算机软件保护条例》第二条的规定，著作权法保护的是计算机程序及其有关文档。

(1) 计算机程序。

根据《计算机软件保护条例》第三条第一款的规定，计算机程序是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列，或者可被自动转换

成代码化指令序列的符号化语句序列。计算机程序包括源程序和目标程序，同一程序的源程序文本和目标程序文本视为同一软件作品。

(2) 计算机程序的文档。

根据《计算机软件保护条例》第三条第二款的规定，计算机程序的文档是指用自然语言或者形式化语言所编写的文字资料和图表，用来描述程序的内容、组成、设计、功能规格、开发情况、测试结果及使用方法等。文档一般以程序设计说明书、流程图和用户手册等表现。

2. 计算机软件受著作权法保护的条件

计算机软件受著作权法保护应符合以下条件。

- 独立创作。
- 可被感知。
- 逻辑合理。

3. 计算机软件著作的权利

1) 计算机软件的著作人身权

计算机软件享有两种权利：人身权(精神权利)和财产权(经济权利)。软件著作人还享有发表权和开发者身份权。

发表权是指是否公布软件作品的权利；开发者身份权又称为署名权，指软件作者在作品中署自己名字的权利。

2) 计算机软件的著作财产权

著作财产权是指能够给著作权人带来经济利益的权利。通常是指由软件著作权人控制和支配，并能够为权利人带来一定经济效益的权利。主要内容有使用权、复制权、修改权、发行权、翻译权、注释权、信息网络传播权、出租权、使用许可权和获得报酬权、转让权。

3) 软件合法持有人的权利

软件合法持有人的权利主要有：根据使用的需要把软件装入计算机等装置内；根据需要进行必要的复制；为了防止复制品损坏而制作备份复制品；为了把该软件用于实际的计算机应用环境而做的必要修改，但不得向第三方提供修改后的软件。

4) 计算机软件著作权的行使

(1) 软件经济权利的许可使用。

软件经济权利的许可使用是指软件著作权人通过合同方式许可他人使用其软件，并获得一定报酬的软件贸易形式。主要有：独占许可使用，被授权方按合同规定取得软件使用的独占性，权利人不得将使用权授予第三方，自己也不得使用该软件；独家许可使用，权利人自己可以使用该软件，其他和独占许可使用相同；普通许可使用，权利人可以将使用权授予第三方，自己也可以使用；法定许可使用和强制许可使用，根据法律特殊规定，不经软件著作权人许可也可以使用其软件。

(2) 软件经济权利的转让使用。

软件经济权利的转让使用是指软件著作权人将其著作权中的经济权利全部转移给他人，受让者成为新的著作权主体。软件著作权的转让必须签订书面合同，同时转让不改变

软件的保护期。转让方式包括卖出、赠与、抵押、赔偿等。

5) 计算机软件著作权的保护期

计算机软件著作权自软件开发完成之日起，保护期为 50 年。保护期满，除开发者身份权外，其他权利终止。计算机软件著作权人的单位终止和计算机软件著作权人的公民死亡无合法继承人时，除开发者身份权外的其他权利进入公有领域。

4. 计算机软件著作权的归属

我国《著作权法》规定著作权属于作者。《计算机软件保护条例》规定软件著作权属于软件开发者。

1) 职务开发软件著作权的归属

公民为完成法人或者其他组织工作任务所创作的作品是职务作品，著作权由作者享有，但法人或者其他组织有权在其业务范围内优先使用。作品完成两年内，未经单位同意，作者不得许可第三人以与单位使用的相同方式使用该作品。

有下列情形之一的职务作品，作者享有署名权，著作权的其他权利由法人或者其他组织享有，法人或者其他组织可以给予作者奖励。

(1) 主要是利用法人或者其他组织的物质技术条件创作，并由法人或者其他组织承担责任的工程设计图、产品设计图、地图、计算机软件等职务作品。

(2) 法律、行政法规规定或者合同约定著作权由法人或者其他组织享有的职务作品。

2) 合作开发软件著作权的归属

合作开发软件是指两个或两个以上公民、法人或其他组织订立协议，共同参加某项计算机软件的开发并分享软件著作权的形式。对合作开发软件著作权的归属应掌握以下四点。

(1) 由两个以上的单位、公民共同开发完成的软件属于合作开发的软件。

(2) 由于合作开发软件著作权是由两个以上单位或者个人共同享有，因而为了避免在软件著作权的行使中产生纠纷，规定“合作开发的软件，其著作权的归属由合作开发者签订书面合同约定”。

(3) 对于合作开发的软件著作权按以下规定执行：“无书面合同或者合同未作明确约定，合作开发的软件可以分割使用的，开发者对各自开发的部分可以单独享有著作权；但是，行使著作权时，不得扩展到合作开发的软件整体的著作权。”

(4) 合作开发者对于软件著作权中的转让权不得单独行使。

3) 委托开发软件著作权的归属

受委托创作的作品，著作权的归属由委托人和受托人通过合同约定。合同未作明确约定或者没有订立合同的，著作权属于受托人。委托开发的软件著作权的归属按以下标准确定。

(1) 委托开发软件作品是根据委托方的要求，由委托方与受托方以合同确定的权利和义务的关系而进行开发的软件。因此，软件作品著作权归属应当作为合同的重要条款予以明确约定。

(2) 若在委托开发软件活动中，委托者与受委托者没有签订书面协议，或者在协议中未对软件著作权归属做出明确的约定，则软件著作权属于受委托者，即属于实际完成软件的开发者。

4)　接受任务开发软件的著作权归属

接受任务开发软件的著作权归属一般按以下两条标准确定：①在合同中明确约定的，按照合同约定实行；②未明确约定的，著作权属于实际完成软件开发的单位。

5)　计算机软件著作权主体变更后软件著作权的归属

因主体变更引起的变化有以下几种。

(1)　公民继承的软件权利归属。合法继承人享有除署名权外的其他权利，例如著作权的使用权、使用许可权和获得报酬权等权利。

(2)　单位变更后软件权利归属。著作权属于法人或者其他组织的，法人或者其他组织变更、终止后，由承受其权利义务的法人或者其他组织享有；没有承受其权利义务的法人或者其他组织的，由国家享有。

(3)　权利转让后的软件著作权归属。权利转让根据签订的合同规定各方的权利。

(4)　司法判决、裁定引起的软件著作权归属问题根据法律的判决来执行。

(5)　保护期届满权利丧失。

5. 计算机软件著作权侵权的鉴别

计算机软件著作权的侵权行为主要有：未经软件著作权人的同意而发表或者登记其作品；将他人开发的软件当作自己的作品发表或者登记；未经合作者同意将与他人合作开发的软件当作自己独立完成的作品发表或者登记；在他人开发的软件上署名或者更改他人开发的软件上的署名；未经软件著作权人或者其合法受让者的许可，修改或翻译其软件作品；未经软件著作权人或其合法受让者的许可，复制或部分复制其软件；未经软件著作权人或其合法受让者的同意，向公众发行出租其软件的复制品；未经软件著作权人或其合法受让者的同意，向任何第三方办理软件权利许可或转让事宜；未经软件著作权人或其合法受让者的同意，通过信息网络传播著作权人的软件；共同侵权，两人以上共同实施的侵权行为。

6. 计算机软件著作权侵权的法律责任

计算机软件著作权侵权的法律责任主要有民事责任、行政责任和刑事责任。

需要承担民事责任的侵权行为有：未经软件著作权人的许可发表或登记其软件的；将他人的软件当作自己的软件发表或登记的；未经合作者许可，将与他人合作开发的软件当作自己独立完成的作品发表或者登记的；在他人开发的软件上署名或者更改他人开发的软件上的署名的；未经软件著作权人或者其合法受让者的许可，修改或翻译其软件的；其他侵犯软件著作权的行为。

需要承担行政责任的侵权行为有：复制或部分复制著作权人软件的；向公众发行、出租著作权人软件的；故意避开或者破坏著作权人为保护其软件而采取的技术措施的；故意删除或者改变软件权利管理电子信息的；许可他人行使或者转让著作权人的软件著作权的。

侵权行为构成犯罪的，侵权者应承担相应的刑事责任。

真题链接

【例 11-2】甲公司接受乙公司委托开发了一项应用软件，双方没有订立任何书面合同，在此情形下，<u>(10)</u>享有该软件的著作权。(2014 年 11 月真题 10)

(10) A. 甲公司　　B. 甲、乙公司共同　　C. 乙公司　　D. 甲．乙公司均不

【解析】接受任务开发软件的著作权归属一般按以下两条标准确定：①在合同中明确约定的，按照合同约定实行；②未明确约定的，著作权属于实际完成软件开发的单位。

【答案】A

【例 11-3】以下作品中，不适用或不受著作权法保护的是__(14)__。(2013 年 11 月真题 14)

(14) A．某教师在课堂上的讲课

B．某作家的作品《红河谷》

C．最高人民法院组织编写的《行政诉讼案例选编》

D．国务院颁布的《计算机软件保护条例》

【解析】对于法律、法规，国家机关的决议、决定、命令和其他具有立法、行政、司法性质的文件，及其官方正式译文，时事新闻不适用著作权法保护。显然，国务院颁布的《计算机软件保护条例》属于政策法规，不适用著作权法保护。

【答案】D

11.2.3 计算机软件的商业秘密权

1. 商业秘密

1) 商业秘密的定义

在我国的《反不正当竞争法》中规定商业秘密是“不为公众所熟悉的、能为权利人带来经济效益、具有实用性并经权利人采取保密措施的技术信息和经营信息”，主要包括经营秘密和技术秘密。

商业秘密权作为一种无形财产权受到法律的保护，在计算机软件中，包含商业秘密的可作为商业秘密权的保护对象。

2) 商业秘密的构成条件

商业秘密的构成条件是：商业秘密必须具有未公开性，即不为公众所知悉；商业秘密必须具有实用性，即能为权利人带来经济效益；商业秘密必须具有保密性，即采取了保密措施。

3) 商业秘密权

商业秘密是一种无形的信息财产。与有形财产相区别，商业秘密不占据空间，不易被权利人所控制，不发生有形损耗，其权利是一种无形财产权。商业秘密的权利人与有形财产所有权人一样，依法享有占有、使用和收益的权利，即有权对商业秘密进行控制与管理，防止他人采取不正当手段获取与使用；有权依法使用自己的商业秘密，而不受他人干涉；有权通过自己使用或者许可他人使用以至转让所有权，从而取得相应的经济利益；有权处理自己的商业秘密，包括放弃占有、无偿公开、赠与或转让等。

4) 商业秘密的丧失

一项商业秘密受到法律保护的依据，必须具备上述构成商业秘密的三个条件，缺少上述三个条件之一就会造成商业秘密丧失保护。

2. 计算机软件商业秘密的侵权

侵犯商业秘密是指未经权利人的许可，以非法手段获得商业秘密并加以公开或使用的行为。其具体行为主要有如下几种。

(1) 以盗窃、利诱、胁迫或以其他不正当手段获取权利人的计算机软件商业秘密。

(2) 披露、使用或允许他人使用以不正当手段获取的计算机软件商业秘密。

(3) 违反约定或违反权利人有关保守商业秘密的要求，披露、使用或允许他人使用其掌握的计算机软件商业秘密。

(4) 第三方在明知前述违法行为的情况下，仍然从侵权人那里获取或使用他人计算机软件商业秘密。该行为属于间接侵权。

3. 计算机软件商业秘密侵权的法律责任

下面介绍计算机软件商业秘密侵权的法律责任。

(1) 侵权者的行政责任：责令其停止侵权行为，处以 1 万元以上 20 万元以下的罚款。

(2) 侵权者的民事责任：侵权人承担损害赔偿责任，被侵权人可以向法院提起诉讼。

(3) 侵权者的刑事责任：侵权者的侵权行为造成重大损害的，侵权者承担刑事责任。

11.2.4　专利权

1. 专利权的保护对象与特征

专利权，是指由国务院专利行政部门授予的，发明创造者在规定的时间内享有的独占使用权，在这一规定的时间内，任何自然人、法人、其他组织，未经其许可，均不得制造其发明创造。依《中华人民共和国专利法》规定，我国国务院专利行政管理部门授予的专利有以下三种：发明专利、实用新型专利、外观设计专利。不属于专利权范围的有如下几种。

- 违反国家法律、社会公德或者妨碍公共利益的发明创造。
- 科学发现。
- 智力活动的规则和方法，如推理、分析、判断、处理等思维活动的方法。
- 疾病诊断手段和治疗方法。
- 动、植物品种。
- 用原子核变换方法获得的物质。

2. 授予专利的条件

授予专利的条件是指发明创造获得专利的实质条件，包括新颖性、创造性和实用性三方面。

(1) 新颖性是指发明创造在申请日之前未被公开也没有同样的发明被申请的。新颖性是创造性的前提。

(2) 创造性是指发明创造要有实质性特点和显著的进步。

(3) 实用性是指发明创造要能够使用，并且能够产生积极的效果。申请专利的技术方案违背自然规律或利用独一无二的自然条件完成的，不具备实用性。

外观设计获得专利权的条件为新颖性和美观性。新颖性和上述相同，美观性是指该设计可以使产品产生美感。

3. 专利的申请

专利申请权是公民、法人或其他组织依据法律规定或者合同约定享有的就发明创造向专利局提出专利申请的权利。专利申请权可以转让、继承或赠与。

专利申请采用书面形式，一项专利申请文件只能申请一项专利。两个或两个以上的人就同样的发明创造申请专利的，专利权授予最先申请人。两个以上的申请人在同一日分别就同样的发明创造申请专利的，应在收到专利行政部门的通知后自行协商确定申请人。

专利申请日，又称关键日，是指专利主管部门收到完整的专利申请文件的日期。如果专利申请文件是邮寄的，以寄出的邮戳日为申请日。

对发明专利的审批要通过实质审查，即依法审查专利的新颖性、创造性和实用性。对实用新型和外观设计专利申请只进行初步审查，不进行实质审查。

申请人在法定期间或专利局指定的期限内未办理相关的手续或未能提供有关文件的，其申请将被撤回，丧失其申请权。

4. 专利权行使

1) 专利权的归属

依《中华人民共和国专利法》及其实施细则的规定，专利权归下列人所有。

(1) 职务发明创造的专利申请权和专利权人为单位。

(2) 非职务发明创造的专利申请权和专利权人为个人。

(3) 利用本单位的物质技术条件所完成的发明创造，其专利申请权和专利权人依其合同约定决定。

(4) 两个以上单位或者个人合作完成的发明创造，除各方在协议中约定的以外，其专利申请权和专利权人属于完成或者共同完成的单位或者个人。

(5) 一个单位或者个人接受其他单位或者个人的委托完成的发明创造，除委托书中有约定的外，其专利申请权和专利权人属于完成或者共同完成的单位或者个人。

(6) 两个以上的申请人分别就同样的发明创造申请专利的，专利权授予最先申请的人。

(7) 委托开发的专利权根据委托开发的协议中的规定来确定，若未有明确规定，则专利权属于专利完成者。

2) 专利权人的权利

专利权是一种具有财产权属性的独占权以及由其衍生出来的相应处分权。专利权人的权利有独占实施权、转让权、实施许可权、放弃权、标记权等。专利实施许可包括独占许可、独家许可、普通许可和部分许可。

5. 专利权的限制

根据我国《专利法》的规定，发明专利的保护期限为 20 年，实用新型和外观设计专利为 10 年。

专利权因某种法律事实的发生而导致其效力消失的情形称为专利权终止。导致专利权

终止的事实有：保护期限届满；专利权人放弃专利权；专利权人没有按规定交纳年费。

6. 专利侵权行为

专利侵权行为是指在专利保护期内，未经专利权人的许可擅自以营利为目的使用专利的行为。其行为主要有如下几种。

(1) 为生产经营目的的制造、使用、销售其专利产品。或者使用其专利方法以及通过该专利方法获得产品。

(2) 为生产经营目的的制造、销售其外观设计专利产品。

(3) 进口依照其专利方法直接获得的产品。

(4) 产品的包装上标明专利标记和专利号。

(5) 冒充专利产品或专利方法等。

对专利侵权行为，专利权人可以请求专利管理机关处理，也可以请求法院审理。侵犯专利的诉讼时效为两年，自专利权人知道或应当知道侵权之日起算起。

真题链接

【例 11-4】M 软件公司的软件产品注册商标为 M，为确保公司在市场竞争中占据优势，对员工进行了保密约束。此情形下该公司不享有__(10)__。(2012 年 11 月真题 10)

(10) A．商业秘密权　　B．著作权　　C．专利权　　D．商标权

【解析】专利权不能自动取得，必须向专利行政部门提出专利申请，履行专利法规定的专利申请手续并向国家专利行政部门提交必要的申请文件。题目中，M 软件公司没有申请过专利，因此不享有专利权。

【答案】C

11.2.5　商标权

商标权是商标专用权的简称，是指商标主管机关依法授予商标所有人对其注册商标受国家法律保护的专有权。商标注册人依法支配其注册商标并禁止他人侵害的权利，包括商标注册人对其注册商标的排他使用权、收益权、处分权、续展权和禁止他人侵害的权利。商标是用以区别商品和服务不同来源的商业性标志，由文字、图形、字母、数字、三维标志、颜色组合或者上述要素的组合构成。

我国商标权的获得必须履行商标注册程序，而且实行申请在先原则。商标是产业活动中的一种识别标志，所以商标权的作用主要在于维护产业活动中的秩序，与专利权的作用主要在于促进产业的发展不同。

根据《商标法》规定，商标权的有效期为 10 年，自核准注册之日起计算，期满前 6 个月内申请续展，在此期间内未能申请的，可再给予 6 个月的宽展期。续展可无限重复进行，每次续展期为 10 年。

商标权的主要特征如下。

(1) 专有性。商标权的专有性又称为独占性或垄断性，是指注册商标所有人对其注册商标享有专有使用权，其他任何单位及个人非经注册商标所有人的许可，不得使用该注册商标。

(2) 时间性。商标权的时间性也称法定时间性，是指商标权为一种有期限的权利，在有效期限内才受法律保护，超过有效期限，商标权即终止，不再受法律保护。

(3) 地域性。商标权具有严格的地域性，这是由商标权的国内法性质所决定的。

真题链接

【例 11-5】甲、乙软件公司于 2013 年 9 月 12 日就其财务软件产品分别申请“大堂”和“大唐”商标注册，两财务软件相似，且经协商双方均不同意放弃使用其申请注册的商标标识。此情形下，(11)获准注册。(2014 年 11 月真题 11)

(11) A. “大堂” B. “大堂”与“大唐”都能

C. “大唐” D. 由甲、乙抽签结果确定谁能

【解析】本案例中，两个商标发音和功能近似，存在侵权风险，根据软件知识产权管理法案，只能有一个获得注册，抽签决定。

【答案】D

11.2.6 企业知识产权的保护

1. 知识产权的保护和利用

目前计算机技术和软件技术的知识产权保护以《著作权法》为主，《专利法》、《商标法》、《反不正当竞争法》、《合同法》为辅。例如，源程序及设计文档作为软件的表现形式受《著作权法》保护，同时作为技术秘密又受《反不正当竞争法》的保护。对企业来说，不能只依靠法律法规，要建立自己的知识产权保护措施，一般可采取以下方法。

(1) 明确软件知识产权归属。

(2) 及时对软件技术秘密采取保护措施。一旦发生泄露，要追究行为人的责任，保护企业权益。

(3) 依靠专利保护新技术和新产品。要及时申请专利，不能拖延导致新成果新颖性的丧失。

(4) 软件产品要尽快完成商标或服务标记的注册，保护产品的商标专有权。

(5) 软件产品进入市场前进行软件著作权登记。

2. 建立经济约束机制规范调整各种关系

软件企业需要建立企业内部以及企业与外部的各种经济约束机制。目前，软件企业应建立以下各项合同规范。

(1) 劳动关系合同。

(2) 软件开发合同。约定软件开发各方享有的权利和义务。

(3) 软件许可使用(或转让)合同。

11.3 应试加油站

11.3.1 考频统计

通过对近几年考试真题的分析，“标准化和软件知识产权基础”模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 11-1 所示。

表 11-1 历年考点统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：10～11	软件著作权归属、知识产权	2
	下午：无		0
2014 年 5 月	上午：10～11	著作权归属、商标注册权	2
	下午：无		0
2013 年 11 月	上午：13～14	著作权及其保护的客体	3
	下午：无		0
2013 年 5 月	上午：10～11	著作权归属、著作权侵权	2
	下午：无		0
2012 年 11 月	上午：10～11	知识产权、专利权	2
	下午：无		0
2012 年 5 月	上午：10～11	著作权、专利权	2
	下午：无		0

纵观历年试卷，本章知识点是以选择题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量大约为 2 道选择题，所占分值为 2 分(约占试卷总分值 75 分中的 3%)。本章试题主要考查对基本概念的理解，难度不高。标准化很少考到，要重点掌握的是知识产权部分。

11.3.2 解题技巧

【典型题 11-1】甲、乙两软件公司于 2012 年 7 月 12 日就其财务软件产品分别申请“用友”和“用有”商标注册。两财务软件相似，甲第一次使用时间为 2009 年 7 月，乙第一次使用时间为 2009 年 5 月。此情形下，(11) 能获准注册。(2014 年 5 月真题 11)

(11) A. “用友”　　B. “用友”与“用有”都

C. “用有”　　D. 由 甲、乙抽签结果确定

【解析】2014 年修订的《中华人民共和国商标法实施条例》第十九条规定“两个或者两个以上的申请人，在同一种商品或者类似商品上，分别以相同或者近似的商标在同一天申请注册的，各申请人应当自收到商标局通知之日起 30 日内提交其申请注册前在先使用该商标的证据。同日使用或者均未使用的，各申请人可以自收到商标局通知之日起 30 日内自行协商，并将书面协议报送商标局；不愿协商或者协商不成的，商标局通知各申请人以抽签的方式确定一个申请人，驳回其他人的注册申请。商标局已经通知但申请

人未参加抽签的，视为放弃申请，商标局应当书面通知未参加抽签的申请人。”

【答案】C

【典型题 11-2】王某是一名软件设计师，按公司规定编写软件文档，并上交公司存档。这些软件文档属于职务作品，且_(10)_。(2013 年 5 月真题 10)

(10) A. 其著作权由公司享有

B. 其著作权由软件设计师享有

C. 除其署名权以外，著作权的其他权利由软件设计师享有

D. 其著作权由公司和软件设计师共同享有

【解析】执行本单位的任务或者主要是利用本单位的物质条件所完成的职务发明创造，著作的权利属于该单位。王某所编写的文档是按公司规定编写的，因此著作权应归属该公司。

【答案】A

【典型题 11-3】甲经销商擅自复制并销售乙公司开发的 OA 软件光盘已构成侵权。丙企业在未知的情形下从甲经销商处购入 10 张并已安装使用。在丙企业知道了所使用的软件为侵权复制品的情形下，以下说法正确的是_(11)_。(2013 年 5 月真题 11)

(11) A. 丙企业的使用行为为侵权，须承担赔偿责任

B. 丙企业的使用行为不侵权，可以继续使用这 10 张软件光盘

C. 丙企业的使用行为侵权，支付合理费用后可以继续使用这 10 张软件光盘

D. 丙企业的使用行为为不侵权，不需要承担任何法律责任

【解析】丙企业是在未知的情况下从甲处购入 10 张光盘，主观上不是故意的，因此不需要承担赔偿责任，但丙购买的是盗版软件，构成侵权。

【答案】C

【典型题 11-4】为说明某一问题，在学术论文中需要引用某些资料。以下叙述中错误的是__(13)__。(2013 年 11 月真题 13)

(13) A．既可引用发表的作品，也可引用未发表的作品

B．只能限于介绍、评论作品

C．只要不构成自己作品的主要部分，可适当引用资料

D．不必征得原作者的同意，不需要向他支付报酬

【解析】为说明某一问题，在学术论文中可以引用已发表的作品，且只限于介绍、评论作品，不能构成自己作品的主要部分。显然，未发表的作品不能引用，故 A 选项错误。

【答案】A

【典型题 11-5】X 软件公司的软件工程师张某兼职于 Y 公司，为完成 Y 科技公司交给的工作，做出了一项涉及计算机程序的发明。张某认为该发明是利用自己的业余时间完成的，可以以个人名义申请专利。此项专利申请权应归属于_(11)_。(2012 年 11 月真题 11)

(11) A．张某　　B．X 软件公司　　C．Y 科技公司　　D．张某和 Y 科技公司

【解析】张某的发明是为了完成 Y 公司交给的工作而产生的结果，很显然该发明为职务发明创造。专利权法规定，职务发明创造的专利申请和获得专利的权利为单位所有。

【答案】C

【典型题 11-6】软件著作权的客体不包括__(10)__。(2012 年 5 月真题 10)

(10) A. 源程序　　B. 目标程序　　C. 软件文档　　D. 软件开发思想

【解析】软件著作权的客体是指著作权法保护的软件著作权的范围，包括计算机程序、软件开发文档，而计算机程序又包括源程序和目标程序。受著作权法保护的作品应当是作者创作思想在固定载体上的一种实际表达，而创作思想未表达出来，不可被感知，就得不到著作权法的保护。

【答案】D

【典型题 11-7】中国企业 M 与美国公司 L 进行技术合作，合同约定 M 使用一项在有效期内的美国专利，但该项美国专利未在中国和其他国家提出申请。对于 M 销售依照该专利生产的产品，以下叙述正确的是__(11)__。(2012 年 5 月真题 11)

(11) A. 在中国销售，M 需要向 L 支付专利许可使用费
B. 返销美国，M 不需要向 L 支付专利许可使用费
C. 在其他国家销售，M 需要向 L 支付专利许可使用费
D. 在中国销售，M 不需要向 L 支付专利许可使用费

【解析】专利权具有合法性、独占性、地域性、时间性和经济性的特征。地域性是指一个国家依照其本国专利法授予的专利权，仅在该国法律管辖的范围内有效，对其他国家没有任何约束力，外国对其专利权不承担保护的义务。题目中涉及的专利是在美国取得的，那么专利权人只在美国享有专有权或独占权。在我国使用该专利权，不属于侵权行为，不需要支付专利许可使用费。

【答案】D

11.4　过 关 习 题

1. 下列关于软件著作权中翻译权的叙述不正确的是：翻译权是指______的权利。
   A. 将原软件从一种自然语言文字转换成另一种自然语言文字
   B. 将原软件从一种程序设计语言转换成另一种程序设计语言
   C. 软件著作权人对其软件享有的以其他各种语言文字形式再表现
   D. 将软件的操作界面或者程序中涉及的语言文字翻译成另一种语言文字

2. ______是指可以不经著作权人许可，不需要支付报酬，使用其作品。
   A. 合理使用　　B. 许可使用　　C. 强制许可使用　　D. 法定许可使用

3. 王某是 M 国际运输有限公司计算机系统管理员。任职期间，王某根据公司的业务要求开发了“海运出口业务系统”，并由公司使用，随后，王某向国家版权局申请了计算机软件著作权登记，并取得了《计算机软件著作权登记证书》。证书明确软件名称是“海运出口业务系统 V1.0”，著作权人为王某。以下说法中，正确的是______。
   A. 海运出口业务系统 V1.0 的著作权属于王某
   B. 海运出口业务系统 V1.0 的著作权属于 M 公司
   C. 海运出口业务系统 V1.0 的著作权属于王某和 M 公司
   D. 王某获取的软件著作权登记证是不可以撤销的

4. 某软件公司研发的财务软件产品在行业中技术领先，具有很强的市场竞争优势。为确保其软件产品的技术领先及市场竞争优势，公司采取相应的保密措施，以防止软件技术秘密的外泄。并且，还为该软件产品冠以“用友”商标，但未进行商标注册。此情况下，公司仅享有该软件产品的______。
   A. 软件著作权和专利权　　B. 商业秘密权和专利权

C. 软件著作权和商业秘密权　　D. 软件著作权和商标权

5. 李某在某软件公司兼职，为完成该公司交给的工作，做出了一项涉及计算机程序的发明。李某认为该发明是自己利用业余时间完成的，可以个人名义申请专利。关于此项发明的专利申请权应归属______。

A. 李某　　B. 李某所在单位

C. 李某兼职的软件公司　　D. 李某和软件公司约定的一方

6. 软件商标权的权利人是指______ 。

A. 软件商标设计人　　B. 软件商标制作人

C. 软件商标使用人　　D. 软件注册商标所有人

第 12 章

计算机专业英语

本章要点

- 计算机专业词汇
- 专业英语解题技巧

12.1 专业英语试题分析

考核说明：专业英语是必考题，分值固定为 5 分，难度不大，这种题目基本都是送分题。最近几年，软件设计师考试中的专业英语都偏向于软件工程方面，因此建议多阅读相关的英文资料。

12.1.1 软件工程专业词汇

abstraction 抽象
access modifier 存取权限
accessor methods 存取器方法
acceptance 验收
action 动作
action sequence 动作序列
action state 动作状态
activation 激活
active class 主动类
activity 活动
active object 主动对象
activity graph 活动图
actor 主角
actor class 主角类
actor-generalization 主角泛化关系
actual parameter 实参
aggregate class 聚合类
aggregation 聚合关系
analysis 分析
analysis class 分析类
analysis & design 分析设计
analysis mechanism 分析机制
analysis pattern 分析模式
analyst 分析员
Application Programming Interface(API) 应用程序编程接口
appraisal 评估
architectural baseline 构架基线
architectural mechanism 构架机制
architectural pattern 构架模式
architectural view 构架视图
architecture 构架
artifact 工件
artifact guidelines 工件指南
artifact set 工件集
association 关联关系
association class 关联类
association end 关联关系端
asynchronous action 异步动作
asynchronous review 异步评审
attribute 属性
base class 基类
baseline 基线
bean 可用于构建应用程序的小构件
behaviour 行为
behavioral feature 行为特性
behavioral model aspect 模型的行为侧重面
beta testing 测试
binary association 二元关联关系
binding 绑定
boundary class 边界类
break point 断点
business actor 业务主角
business actor class 业务主角类
business creation 业务创建
business engineering 业务工程
business entity 业务实体
business improvement 业务改进
business object model 业务对象模型
business modeling 业务建模
business process 业务过程
business process engineering 业务过程工程

business reengineering 业务重建
business rule 业务规则
business use case 业务用例
business use-case instance 业务用例实例
business use-case model 业务用例模型
business use-case package 业务用例包
business use-case realization 业务用例实现
business worker 业务角色
capsule 封装体
cardinality 基数
causal analysis 因果分析
change management 变更管理
Change Request (CR) 变更请求
checklist 检查表
checkpoints 检查点
class 类
class diagram 类图
class hierarchy 类分层结构
class library 类库
class method 类方法
classifier 分类器
client 客户端
client/server 客户机/服务器
collaboration 协作
collaboration diagram 协作图
comment 注释
component 构件
component diagram 构件图
component model 构件模型
Component-Based Development (CBD) 基于构件的开发
composite aggregation 组装关系
composite class 组装类
composite state 组合状态
composite substate 组合子状态
composition 组装
concrete 具体
concrete class 具体类
concurrency 并行
concurrent substate 并行子状态
configuration 配置
configuration item 配置项
configuration management 配置管理
constraint 约束
construction 构建
constructor 构造函数
container 容器
containment hierarchy 容器分层结构
context 环境
control chart 控制图
control class 控制类
conversational 会话式
Critical Design Review (CDR) 关键设计评审
customer 客户
cycle 周期
database 数据库
datatype 数据类型
decision rule 决策规则
defect 缺陷
defect checklist 缺陷检查表
defect density 缺陷密度
defect log 缺陷日志
defining model 定义模型
delegation 委托
deliverable 可交付工件
de-marshal 串行化
Demilitarized Zone (DMZ) 隔离带
dependency 依赖关系
deployment 部署
deployment diagram 部署图
deployment unit 部署单元
deployment view 部署视图
derived element 派生元素
deserialize 反串行化
design 设计
design mechanism 设计机制
design model 设计模型
design package 设计包
design pattern 设计模式
design subsystem 设计子系统
developer 开发人员
development case 开发案例

development process 开发过程
device 设备
diagram 图
disjoint substate 互斥子状态
distributed processing 分布式处理
document 文档
document description 文档说明
document template 文档模板
domain 领域
dynamic classification 动态分类
dynamic information 动态信息
elaboration 精化
element 元素
encapsulation 封装
enclosed document 附带文档
enhancement request 扩展请求
entity class 实体类
entry action 进入动作
error 错误
event 事件
event-to-method connection 事件-方法映射
evolution 演进
evolutionary 演进方式
executable architecture 可执行构架
exit action 退出动作
exit criteria 准出条件
export 导出
expression 表达式
extend 扩展
extend-relationship 扩展关系
facade 外观
factory 工厂
fault 故障
feature 特性
field 字段
final state 最终状态
focus of control 控制焦点
follow-up 跟踪
formal review 正式评审
formal parameter 形参
framework 框架
gateway 网关
generalizable element 可泛化元素
generalization 泛化关系
generation 代
green-field development 零起点开发
guard condition 警戒条件
hash table 哈希表
header file 头文件
heap 堆
idiom 代码模式
implementation 实施
implementation inheritance 实施继承
implementation mechanism 实施机制
implementation model 实施模型
implementation pattern 实施模式
implementation subsystem 实施子系统
implementation view 实施视图
import 导入
import-dependency 导入依赖关系
inception 先启
include 包含
include-relationship 包含关系
increment 增量
incremental 递增
informal review 非正式评审
inheritance 继承
injection rate 缺陷率
input 输入
inspection 审查
inspection effectiveness 审查有效性
inspection efficiency 审查效率
inspection package 审查包
inspection summary report 审查总结报告
inspector 审查者
issue 问题
issue log 问题日志
instance 实例
Integrated Development Environment (IDE) 集成开发环境
integration 集成
integration build plan 集成构建计划

interaction 交互
interaction diagram 交互图
interface 接口/界面
interface inheritance 接口继承
internal transition 内部转移
iteration 迭代
Java Archive(JAR) Java 档案文件
Java Development Kit(JDK) Java 开发工具包
Java Foundation Classes(JFC) Java 基础类
key mechanism 关键机制
keyword 关键字
link 链接
link end 链接端
listener 监听程序
logical view 逻辑视图
major defect 主要缺陷
management 管理
marshal 反串行化
measurement dysfunction 测量混乱
mechanism 机制
message 消息
messaging 消息传递
metaclass 元类
meta-metamodel 元-元模型
metamodel 元模型
metaobject 元对象
method 方法
method call 方法调用
metric 度量
milestone 里程碑
minor defect 次要缺陷
model 模型
model aspect 模型侧重面
model elaboration 模型精化
model element 模型元素
Model View Controller (MVC) 模型视图控制器
modeling conventions 建模约定
moderator 评审组长
module 模块
multiple classification 多重分类
multiple inheritance 多重继承
multiplicity 多重性
multi-valued 多值
mutator methods 存取器方法
n-ary association 多元关联关系
n-fold inspection N重审查
namespace 名字空间
node 结点
object 对象
object class 对象类
object diagram 对象图
object flow state 对象流状态
object lifeline 对象生命线
object model 对象模型
Object Request Broker (ORB) 对象请求代理
Object-Oriented Programming(OOP)
面向对象程序设计
operation 操作
organization unit 组织单元
originator 发起者
output 输出
outside link 外部链接
package 包
pair programming 结对编程
palette 调色板
parameter 参数
parameter connection 参数连接
parameterized element 参数化元素
parent 父
parent class 父类
participates 参与
partition 分区
passaround 轮查
pattern 模式
peer deskcheck 同级桌查
peer review 同级评审
peer review coordinator 同级评审协调者
persistent object 永久对象
phase 阶段
port 端口
post-condition 后置条件
pre-condition 前置条件

Preliminary Design Review(PDR) 初步设计评审
primitive type 基础类型
private 私有
process 进程、过程
process assets library 过程资产库
process owner 过程拥有者
process view 进程视图
processor 处理器
product 产品
product champion 产品推介人
product-line architecture 产品线构架
Product Requirements Document(PRD) 产品需求文档
project 项目
project manager 项目经理
Project Review Authority(PRA) 项目评审委员会
projection 投影
promotion 晋升
property 特征
property-to-property connection 特征-特征连接
protected 保护
protocol 协议
prototype 原型
proxy 代理
pseudo-state 伪状态
published model 已发布的模型
qualifier 限定词
Quality Assurance(QA) 质量保证
race condition 竞争状态
rank 等级
rationale 理由
receive a message 接收消息
receiver object 接收方对象
reception 接收
reference 引用
refinement 改进
relationship 关系
release 发布版
release manager 发布经理
Remote Method Invocation(RMI) 远程方法调用
Remote Procedure Call(RPC) 远程过程调用
report 报告
repository 储存库
requirement 需求
requirement attribute 需求属性
requirements 需求
requirements management 需求管理
requirements tracing 需求跟踪
requirement type 需求类型
resource file 资源文件
responsibility 职责
result 结果
resurrect 反串行化
review 评审
reuse 复用
rework 返工
risk 风险
role 角色
rule 规则
sandbox 沙箱
scenario 场景
scope management 范围管理
semantic variation point 语义分歧点
send a message 发送消息
sender object 发送方对象
sequence diagram 序列图
serialize 串行化
severity 严重性
signal 信号
signature 签名
single inheritance 单重继承
single valued 单值
single-byte character set 单字节字符集
socket secure 套接字保护
software architecture 软件构架
Software Engineering Process Group(SEPG) 软件工程过程组
software requirement 软件需求
Software Requirements Specifications(SRS) 软件需求规约
Software Specification Review(SSR) 软件规约评审

specification 规约
stakeholder 涉众
stakeholder need 涉众需要
stakeholder request 涉众请求
start page 起始页
state 状态
statechart diagram 状态图
static artifact 静态工件
static classification 静态分类
static information 静态信息
stereotype 构造型
stimulus 激励
structural feature 结构特性
structural model aspect 模型的结构侧重面
stub 桩模块
subactivity state 子活动状态
subclass 子类
submachine state 子机状态
substate 子状态
subsystem 子系统
subtype 子类型
superclass 超类
supertype 超类型
supplier 提供端
swimlane 泳道
system 系统
System Requirements Review(SRR) 系统需求评审
tagged value 标注值
task 任务
team leader 团队负责人
technical authority 技术权威
template 模板
test 测试
test case 测试用例
test coverage 测试覆盖
test driver 测试驱动程序
test item 测试项
test procedure 测试过程
thread 线程
time event 时间事件
time expression 时间表达式
timing mark 时间标记
tool mentor 工具向导
traceability 可追踪性
trace 追踪
transaction 事务
transaction processing 事务处理
transient object 临时对象
transition 产品化/转移
type 类型
type expression 类型表达式
typo list 微错清单
unicode 统一编码
Unified Modeling Language(UML) 统一建模语言
Uniform Resource Locator(URL) 统一资源定位符
usage 用途
use case 用例
use-case diagram 用例图
use-case instance 用例实例
use-case model 用例模型
use-case package 用例包
use-case realization 用例实现
use-case view 用例视图
User Interface (UI) 用户界面
utility 实用工具
validation 确认
value 值
variable 变量
verification 验证
version 版本
vertex 顶点
view 视图
view element 视图元素
view projection 视图投影
Virtual Machine(VM) 虚拟机
visibility 可见性
vision 前景
visual programming tool 可视化编程工具
walkthrough 走查
work breakdown structure 工作细分结构
work guideline 工作指南
work product 工作产品
worker 角色
workflow 工作流程
workflow detail 工作流程明细

真题链接

【例 12-1】(2014 年 11 月真题 71～75)

Teams are required for most engineering projects. Although some small hardware or software products can be developed by individuals, the scale and complexity of modem systems is such, and the demand for short schedules so great, that it is no longer (71) for one person to do most engineering jobs. Systems development is a team (72) , and the effectiveness of the team largely determines the (73) of the engineering.

Development teams often behave much like baseball or basketball teams. Even though they may have multiple specialties, all the members work toward (74) .However, on systems maintenance and enhancement teams, the engineers often work relatively independently, much like wrestling and track teams.

A team is (75) just a group of people who happen to work together. Team work takes practice and it involves special skills. Teams require common processes; they need agreed-upon goals; and they need effective guidance and leadership. The methods for guiding and leading such teams are well known, but they are not obvious.

(71) A. convenient　B. existing　C. practical　D. real

(72) A. activity　B. job　C. process　D. application

(73) A. size　B. quality　C. scale　D. complexity

(74) A. multiple objectives　B. different objectives　C. a single objective　D. independent objectives

(75) A. relatively　B. /　C. only　D. more than

【解析】大多数的项目工程需要团队。虽然一些小的硬件或软件产品可以由个人开发，就项目的规模和系统的复杂性以及对时间短的要求，一个人做大多数项目工作是不实际的。系统开发是一个团队活动，团队效率在很大程度上决定了的项目的质量。

开发团队往往表现得像棒球或篮球队。尽管他们可能有多个专业，但所有成员的努力都是为了一个目标。然而，从对系统的维护和团队的增强方面考虑，工程师们的工作又是相对独立的，就像摔跤和田径队。

一个团队不仅仅只是一群人在一起工作。团队协作需要实践，它涉及很多特殊技能。团队需要共同的进步；他们需要达成共同的目标；他们需要有效的指导和领导。指导和领导团队的方法是众所周知的，但他们并不明显。

【答案】(71)C；(72)A；(73)B；(74)C；(75)D

12.1.2　专业英语试题分析

计算机专业英语一般是以完形填空的形式出现在 71～75 题。具体而言，完形填空主要考查应试者对语篇中句法、词语和短语的把握能力，具有较强的测试性。每一个空都要通过上下文进行综合考虑，仅仅依靠一个单句往往无法确立正确选项。

语篇的内容往往是对软件设计中相关知识的描述，需要应试者对这些内容有一定的了解。

1. 完形填空中的句法

计算机英语的完形填空，句法强调时态、语态、倒装、复合，同时要求主语、谓语和宾语结构在数、格等方面的一致性。此外，连接手段包括关系代词、关系副词连接词等，

要求与整个语篇的行文相一致，起到或承接、或转折、或加强的作用，有着非常突出的个性特征。

在描述某项事务的发展历史时，一般采用过去时态；对目前还在使用中的技术，采取完成时态或现在时；对未来技术的展望，大都采用将来时。句中几个受同一时间状态限制的动词时态在表达形式上要保持一致。这里包括并列的谓语动词以及主句和从句中谓语动词在表达形式上的一致。

计算机英语的语篇在描述技术类知识时，语态一般力求客观，采用描述性和被动语态较多。这里要注意只有及物动词及相当于及物动词的词组才有被动语态的表达形式。在并列结构中，同样的语义往往需要同样的语态表达形式。

2. 完形填空中的短语和固定用法

英语中有相当数量的动词短语、介词短语和固定搭配，其来源广泛，搭配方式丰富多变。因此需要应试者从动词入手，熟悉固定搭配，尤其是动词短语；从介词入手，了解介词本身的意义，进而了解同一个介词与不同动词、名词搭配产生的不同或相关的意义；理解固定搭配的外延，增强对语义提示的审查力。

3. 完形填空的答题要领

关于完形填空要注意如下答题要领。

(1) 通过首句或出现的核心词汇来推断全文的信息。短文的首句往往是主题句，或出现了核心词汇，能为理解文章的大意和主要内容提供必要线索。一般首句还提供背景资料，因此要特别注意首句，抓住整个段落的纲要。

(2) 把握文章发展的基本线索。文章总是按照一定思路发展起来的，不同的逻辑关系主要依靠使用逻辑连接词来表达，文章如果没有出现内在的逻辑关系，就会语义不清、逻辑混乱。所以通过表示逻辑关系的词汇把握文章发展的基本线索是至关重要的。

借助语法知识和专业背景知识确定正确的词汇选项。

计算机专业英语词汇的考查在试题中占一定比例，词汇选项的设计和文章难度的制定与语法都息息相关。应试者务必借助语法知识和专业背景知识来确定正确的词汇选项。同时注意填入的词汇和文中句子的结构要相一致。

4. 完形填空的答题步骤

关于完形填空题可采取以下答题步骤。

(1) 通读全文。完形填空是考查在全面理解内容的基础上运用语言的能力，由于试题篇幅较短，完全有时间利用通读对全文内容有一个基本的了解。应试者要快速阅读段落，把握基本观点，通读时以浏览为主，可以忽略细节。

(2) 复读答题。在通读的基础上，应试者最好能立即复读，并结合选项，从语法结构、语义、词义、固定搭配等方面结合专业知识来考虑选项。选定之后，还需要回读。在整个答题过程中，切记全文的整体意义，保持思路的连贯性，从而做出正确选择。

(3) 重读检查。在确定所有选项以后，一定要重读全文，检查并核实每个选项在整篇文章中没有造成语义、结构、逻辑等方面的差错，确保短文是一个内容连贯、层次清晰、中心思想突出的整体。

真题链接

【例 12-2】(2014 年 5 月真题 71～75)

Cloud computing is a phrase used to describe a variety of computing concepts that involve a large number of computers __(71)__ through a real-time communication network such as the Internet. In science, cloud computing is a __(72)__ for distributed computing over a network, and means the __(73)__ to run a program or application on many connected computers at the same time.

The architecture of a cloud is developed at three layers: infrastructure, platform, and application.The infrastucture layer is built with virtualized compute storage and network resources. The platform layer is for general-purpose and repeated usage of the collection of software resources. The application layer is formed with a collection of all needed software modules for SaaS applications. The infrastucture layer serves as the __(74)__ for building the platform layer of the cloud. In turn, the platform layer is foundation for implementing the __(75)__ layer for SaaS application.

(71) A.connected B.implemented C.optimized D.virtualized
(72) A.replacement B.switch C.substitute D.synonym(同义词)
(73) A.ability B.approach C.function D.method
(74) A.network B.foundation C.software D.hardware
(75) A.resource B.service C.application D.software

【解析】云计算是一个用来描述各种计算概念的一个短语，涉及大量计算机，这些计算机通过实时通信的网络，如 Internet，连接在一起。在科学研究中，云计算是分布式网络计算的同义词，意味着有能力同时在多台互连的计算机上运行一个程序或应用。

云的结构分为 3 层：基础设施、平台和应用。基础实施层由虚拟计算机存储和网络资源构成；平台层是具有通用性和复用性的软件资源的集合；应用层是云上针对 SaaS 应用的所有应用软件的集合。基础设施层是建立平台层的硬件基础；相应地，平台层是执行应用层 SaaS 应用的基础。

【答案】(71)A；(72)D；(73)A；(74)D；(75)C

12.2 应试加油站

12.2.1 考频统计

通过对近几年考试真题的分析，“计算机专业英语”知识模块在历次软件设计师考试试卷中出现的考核知识点及分值分布情况如表 12-1 所示。

表 12-1 历年考题知识点分布统计表

年 份	题 号	知 识 点	分值/分
2014 年 11 月	上午：71～75	计算机软件设计专业英语	5
	下午：无		0
2014 年 5 月	上午：71～75	计算机软件设计专业英语	5
	下午：无		0

续表

年 份	题 号	知 识 点	分值/分
2013 年 11 月	上午：71～75	计算机软件设计专业英语	5
	下午：无		0
2013 年 5 月	上午：71～75	计算机软件设计专业英语	5
	下午：无		0
2012 年 11 月	上午：71～75	计算机软件设计专业英语	5
	下午：无		0
2012 年 5 月	上午：71～75	计算机软件设计专业英语	5
	下午：无		0

纵观历年试卷，本章知识点是以选择题的形式出现在试卷中的。在历次考试上午试卷中，所考查的题量为 5 道选择题，所占分值为 5(占试卷总分值 75 分中的 6%)。本章考题主要考查应试者结合计算机专业技术知识对全文综合理解的程度和串联上下文的能力、应试者语法知识和对句法结构的辨识能力、应试者的词汇量和词汇运用能力。试题难度不高，应重点了解软件工程方面的专业词汇。

12.2.2　解题技巧

【典型题 12-1】(2013 年 11 月真题 71～75)

There is nothing in this world constant but inconstancy. --SWIFT

Project after project designs a set of algorithms and then plunges into construction of customer- deliverable software on a schedule that demands delivery of the first thing built.

In most projects, the first system built is (71) usable. It may be too slow, too big, awkward to use, or all three. There is no (72) but to start again, smarting but smarter, and build a redesigned version in which these problems are solved. The discard and (73) may be done in one lump, or it may be done piece-by-piece. But all large-system experience shows that it will be done. Where a new system concept or new technology is used, one has to build a system to throw away, for even the best planning is not so omniscient(全知的) as to get it right the first time.

The management question, therefore, is not whether to build a pilot system and throw it away. You will do that. The only question is whether to plan in advance to build a (74) , or to promise to deliver the throwaway to customers. Seen this way, the answer is much clearer. Delivering that throwaway to customers buys time, but it does so only at the (75) of agony(极大痛苦) for the user, distraction for the builders while they do the redesign, and a bad reputation for the product that the best redesign will find hard to live down.

Hence plan to throw dne away; you will, anyhow.

(71) A. almost　B. often　C. usually　D. barely

(72) A. alternative　B. need　C. possibility　D. solution

(73) A. design　B. redesign　C. plan　D. build

(74) A. throwaway　B. system　C. software　D. product

(75) A. worth　B. value　C. cost　D. invaluable

【解析】不变只是愿望，变化才是永恒。——SWIFT

一个接一个的软件项目都是一开始设计算法，然后将算法应用到待发布的软件中，接着根据时间进度把第一次开发的产品发布给客户。

对于大多数项目，第一个开发的系统并不适用。它可能太慢、太大、难以使用，或者三者兼有。要解决所有的问题，除了重新开始以外，没有其他的办法——即开发一个更灵巧或者更好的系统。系统的丢弃和重新设计可以一步完成，也可以一块块地实现。所有开发大型系统的经验都显示，这是必须完成的步骤。而且，新的系统概念或新技术会不断出现，因此开发的系统必须被抛弃，但即使是最优秀的项目计划也不能无所不知地在最开始就解决这些问题。

因此，管理上的问题不再是“是否构建一个实验性的系统，然后抛弃它”，你必须这样做。现在的问题是“是否预先计划抛弃原型的开发，或者是否将该原型发布给用户”。从这个角度看待问题，答案更加清晰。将原型发布给用户，虽然可以获得时间，但是其代价高昂——对于用户，使用极度痛苦；对于重新开发的人员，分散了精力；对于产品，影响了声誉，即使是最好的再设计也难以挽回名声。

因此，为舍弃而计划，无论如何，你一定要这样做。

【答案】(71)D；(72)A；(73)B；(74)A；(75)C

【典型题 12-2】(2013 年 5 月真题 71～75)

So it is today. Schedule disaster, functional misfits,and system bugs all arise because the left hand doesn't know what the right hand is doing. As work (71) , the several teams slowly change the functions, size, and speeds of their own programs, and they explicitly or implicitly (72) their assumptions about me inputs available and the uses to be made of the outputs.

For example, the implementer of a program-overlaying function may run into problems and reduce speed relying on statistics that show how (73) this function will arise in application programs. Meanwhile back at the ranch, his neighbor may be designing a major part of the supervisor so that it critically depends upon the speed of this function. This change in speed itself becomes a major specification change, and it needs to be proclaimed abroad and weighed from a system point of view.

How, then, shall teams (74) with one another? In as many ways as possible.

●Informally. Good telephone service and clear definition of intergroup dependencies will encourage the hundreds of calls upon which common interpretation of written documents depends.

●Meetings. Regular project meetings, with one team after another giving technical briefings, are (75) . Hundreds of minor misunderstandings get smoked out this way.

● Workbook. A formal project workbook must be started at the beginning.

(71) A. starts	B. proceeds	C. stops	D. speeds
(72) A. change	B. proceed	C. smooth	D. hide
(73) A. frequently	B. usually	C. commonly	D. rarely
(74) A. work	B. program	C. communicate	D. talk
(75) A. worthless	B. valueless	C. useless	D. invaluable

【解析】现在，其实也是这样。因为左手不知道右手在做什么，所以进度灾难、功能的不合理和系统缺陷纷纷出现。随着工作的继续进行，许多小组慢慢地修改自己程序的功能、规模和速度，他们明确或者隐含地更改了一些有效输入和输出结果用法上的约定。

例如，程序覆盖功能的实现者遇到的问题，并且统计报告显示了应用程序很少使用该功能。基于这些考虑，他降低了实现覆盖功能的速度。与此同时，这个开发团队中，其他同事可能正在设计监控程序，而监控程序在很大程度上依赖于覆盖功能，它在速度上的变化已成为主要的规格说明变更。因此需要从系统角度来考虑和衡量该变化，以及公开、广泛地发布变更结果。

那么，团队如何进行相互之间的交流沟通呢？通过所有可能的途径。

① 非正式途径。清晰定义小组内部的相互关系，充分利用电话，能鼓励大量的电话沟通，从而达到对所书写文档的共同理解。

② 会议。在常规项目会议中，团队一个接一个地进行简要的技术陈述。这种方式非常有用，能澄清成千上百的细小误解。

③ 工作手册。在项目的开始阶段，应该准备正式的项目工作手册。

【答案】(71)B；(72)A；(73)D；(74)C；(75)D

【典型题 12-3】(2012 年 11 月真题 71～75)

Computer will become more advanced and they will also become easier to use. Improved speed recognition will make the operation of a computer easier. Virtual reality(虚拟现实), the technology of (71) with a computer using all of the human senses, will also contribute to better human and computer (72). Other, exotic(奇异的) models of computation are being developed, including biological computing that uses living organisms, molecular computing that uses molecules with particular (73), and computing that uses DNA, the basic unit of heredity(遗传) to store data and carry out operations. These are examples of possible future computational platforms that, so far, are limited in abilities or are strictly (74). Scientists investigate them because of the physical limitations of miniaturizing circuits embedded in silicon. There are also (75) related to heat generated by even the tiniest of transistors.

(71) A. interact　B. interacting　C. communicate　D. using

(72) A. interfaces　B. behavior　C. similarities　D. comparison

(73) A. software　B. properties　C. programs　D. hardware

(74) A. empirical　B. real　C. practical　D. theoretical

(75) A. developments　B. advantages　C. limitations　D. improvements

【参考翻译】计算机将越来越高级且易于使用。识别速度的提高将使计算机的操作更加容易。虚拟现实是一种用人的所有感官与计算机交互的技术，它也将有助于改善人类和计算机的交互。其他方面，奇异的计算模型都正在开发中，包括使用生物体的生物计算、使用具有特殊性质分子的分子计算及使用 DNA 来存储数据和执行操作的计算。到目前为止，这些可能的未来计算平台都受到性能上的限制或是严格理论化的。科学家研究它们，是因为嵌入在硅中的小型电路存在物理限制，同时存在着产生热量的限制，即使最小的晶体管也会产生热量。

【答案】(71)B；(72)A；(73)B；(74)D；(75)C

【典型题 12-4】(2012 年 5 月真题 71～75)

At a basic level, cloud computing is simply a means of delivering IT resources as (71) . Almost all IT resources can be delivered as a cloud service: applications, compute power, storage capacity, networking, programming tools, even communication services and collaboration (72) .

Cloud computing began as large-scale Internet service providers such as Google, Amazon, and others built

out their infrastructure. An architecture emerged: massively scaled, __(73)__ distributed system resources, abstracted as virtual IT services and managed as continuously configured, pooled resources. In this architecture, the data is mostly resident on __(74)__ "somewhere on the Internet" and the application runs on both the "cloud servers" and the user's browser.

Both clouds and grids are built to scale horizontally very efficiently. Both are built to withstand failures of __(75)__ elements or nodes. Both are charged on a per-use basis. But while grids typically process batch jobs, with a defined start and end point, cloud services can be continuous. What's more, clouds expand the types of resources available-file storage, databases, and Web services-and extend the applicability to Web and enterprise applications.

(71) A. hardware　　B. computers　　C. services　　D. software
(72) A. computers　　B. disks　　C. machines　　D. tools
(73) A. horizontally　　B. vertically　　C. inclined　　D. decreasingly
(74) A. clients　　B. middleware　　C. servers　　D. hard disks
(75) A. entire　　B. individual　　C. general　　D. separate

【参考翻译】基本上，云计算仅仅意味着将 IT 资源作为服务来提供。几乎所有 IT 资源都可以作为一种云服务进行传递，如应用程序、计算能力、存储容量、网络、编程工具，甚至是通信服务和协作工具。

云计算开始被大规模的因特网服务提供商，如谷歌、亚马逊，以及其他一些公司用来构建框架。一种框架开始出现：规模大、系统资源水平分布，抽象成虚拟的 IT 服务，可持续配置管理，资料集中。在这种框架中，数据主要存储在服务器(因特网的某个地方)中，应用程序在云服务器和用户浏览器两端同时运行。

云技术和网格技术的建立都非常有效地扩大了规模，可以经得住单个元素或者节点的失败。两者都受每次基点使用的控制。但是，网格主要处理批量的工作，有一个定义好的起点和终点，而云服务则是连续的。更为重要的是，云技术扩展了可获取资源(文件存储、数据库、Web 服务)的类型，扩展了对 Web 和企业应用的适应性。

【答案】(71)C；(72)D；(73)A；(74)C；(75)B

12.3 过 关 习 题

1. Extreme Programming (XP) is a discipline of software development with __(1)__ of simplicity, communication, feedback and courage. Successful software development is a team effort - not just the development team, but the larger team consisting of customer, management and developers. XP is a simple process that brings these people together and helps them to successed together. XP is aimed primarily at object-oriented projects using teams of a dozen or fewer programmers in one location. The principles of XP apply to any __(2)__ project that needs to deliver quality software rapidly and flexibly.

An XP project needs a __(3)__ customer to provide guidance. Customers, programmers, managers, are all working __(4)__ to build the system that's needed. Customers - those who have software that needs to be developed - will learn simple, effective way to __(5)__ what they need, to be sure that they are getting what they need, and to steer the project to success.

(1) A. importance　　B. keys　　C. roles　　D. values

(2) A. small-sized　　B. moderately-sized
C. large-sized　　D. huge-sized

(3) A. part-time　　B. casual　　C. seldom　　D. full-time

(4) A. together　　B. by themselves　　C. separately　　D. alone

(5) A. tell　　B. know　　C. communicate　　D. feedback

2. Ravi, like many project __(1)__ , had studied the waterfall model of software development as the primary software life-cycle __(2)__ . He has all set to use it for an upcoming project, his first assignment. However, Ravi found that the waterfall model could not be used because the customer wanted the software delivered in stages, something that implied that the system had to be delivered and built in __(3)__ and not as __(4)__ .

The situation in many other projects is not very different. The real world rareley presents a problem in which a standard process, or the process used in a previous project, is the best choice. To be the most situation, an existing process must be __(5)__ to the new problem.

A development process, even after tailoring, generally cannot handle change requests. To accommodate change requests without losing control of the project, you must suppliment the development process with a requirement change management process.

(1) A. customers　　B. managers　　C. users　　D. administrators

(2) A. activity　　B. procedure　　C. process　　D. progress

(3) A. parts　　B. modules　　C. software　　D. a whole

(4) A. parts　　B. modules　　C. software　　D. a whole

(5) A. modified　　B. used　　C. suited　　D. tailored

第 13 章

模拟试卷及参考答案

13.1 模拟试卷

13.1.1 模拟试卷一

上午科目

本试卷共 75 空，每空 1 分，共 75 分。

● (1) 不属于计算机控制器中的部件。

(1) A. 指令寄存器(IR)　　B. 程序计数器(PC)
C. 算术逻辑单元(ALU)　　D. 程序状态字寄存器(PSW)

●若内存按字节编址，用存储容量为 32K× 8 比特的存储器芯片构成地址编号为 A0000H～DFFFFH 的内存空间，则至少需要 (2) 片。

(2) A. 4　　B. 6　　C. 8　　D. 10

●高速缓存 Cache 与主存间采用全相连地址映像方式，高速缓存的容量为 4MB，分为 4 块，每块 1MB，主存容量为 256MB。若主存读写时间为 30ns，高速缓存的读写时间为 3ns，平均读写时间为 3.27ns，则该高速缓存的命中率为 (3) %。若地址变换表如下所示，则主存地址为 8888888H 时，高速缓存地址为 (4) H。

地址变换表

0	38H
1	88H
2	59H
3	67H

(3) A. 90　　B. 95　　C. 97　　D. 99

(4) A. 488888　　B. 388888　　C. 288888　　D. 188888

●若每一条指令都可以分解为取指、分析和执行三步。已知取指时间 $t_{取指}=4\Delta t$，分析时间 $t_{分析}=3\Delta t$，执行时间 $t_{执行}=5\Delta t$。如果按串行方式执行完 100 条指令需要 (5) Δt。如果按照流水方式执行，执行完 100 条指令需要
(6) Δt。

(5) A. 1190　　B. 1195　　C. 1200　　D. 1205

(6) A. 504　　B. 507　　C. 508　　D. 510

●下列行为不属于网络攻击的是 (7) 。

(7) A. 连续不停 ping 某台主机
B. 发送带病毒和木马的电子邮件
C. 向多个邮箱群发一封电子邮件
D. 暴力破解服务器密码

●以下不属于网络安全控制技术的是 (8) 。

(8) A. 防火墙技术　　B. 访问控制技术
C. 入侵检测技术　　D. 差错控制技术

●关于路由器，下列说法中错误的是__(9)__。

(9) A. 路由器可以隔离子网，抑制广播风暴
B. 路由器可以实现网络地址转换
C. 路由器可以提供可靠性不同的多条路由选择
D. 路由器只能实现点对点的传输

●若某人持有盗版软件，但他本人确实不知道该软件是盗版的，则__(10)__承担侵权责任。

(10) A. 应由该软件的持有者　　B. 应由该软件的提供者
C. 应由该软件的提供者和持有者共同　　D. 该软件的提供者和持有者都不

●如果两名以上的申请人分别就同样的发明创造申请专利，专利权应授予__(11)__。

(11) A. 最先发明的人　　B. 最先申请的人　　C. 所有申请人　　D. 协商后的申请人

●对同一段音乐可以选用 MIDI 格式或 WAV 格式来记录存储。以下叙述中__(12)__是不正确的。

(12) A. WAV 格式的音乐数据量比 MIDI 格式的音乐数据量大
B. 记录演唱会实况不能采用 MIDI 格式的音乐数据
C. WAV 格式的音乐数据没有体现音乐的曲谱信息
D. WAV 格式的音乐数据和 MIDI 格式的音乐数据都能记录音乐波形信息

●在彩色喷墨打印机中，将油墨进行混合后得到的颜色称为__(13)__色。

(13) A. 相减　　B. 相加　　C. 互补　　D. 比例

●设计制作一个多媒体地图导航系统，使其能根据用户需求缩放地图并自动搜索路径，最适合的地图数据应该是__(14)__。

(14) A. 真彩色图像　　B. 航拍图像　　C. 矢量化图形　　D. 高清晰灰度图像

●结构化开发方法中，数据流图是__(15)__阶段产生的成果。

(15) A. 需求分析　　B. 总体设计　　C. 详细设计　　D. 程序编码

●以下关于原型化开发方法的叙述中，不正确的是__(16)__。

(16) A. 原型化方法适应于需求不明确的软件开发
B. 开发过程中，可以废弃不用早期构造的软件原型
C. 原型化方法可以直接开发出最终产品
D. 原型化方法利于确认各项系统服务的可用性

●进行软件项目的风险分析时，风险避免、风险监控和风险管理及意外事件计划是__(17)__活动中需要考虑的问题。

(17) A. 风险识别　　B. 风险预测　　C. 风险评估　　D. 风险控制

●软件能力成熟度模型(CMM)将软件能力成熟度自低到高依次划分为初始级、可重复级、定义级、管理级和优化级，并且高级别成熟度一定可以达到低级别成熟度的要求。其中__(18)__中的开发过程及相应的管理工作均已标准化、文档化，并已建立完善的培训制度和专家评审制度。

(18) A. 可重复级和定义级　　B. 定义级和管理级
C. 管理级和优化级　　D. 定义级、管理级和优化级

●选择软件开发工具时，应考虑功能、__(19)__、稳健性、硬件要求和性能、服务和支持。

(19) A. 易用性　　B. 易维护性　　C. 可移植性　　D. 可扩充性

●下面关于编程语言的各种说法中，__(20)__是正确的。

(20) A. 由于 C 语言程序是由函数构成的，因此它是一种函数型语言

B. Smalltalk、C++、Java、C#都是面向对象语言

C. 函数型语言适用于编写处理高速计算的程序，常用于超级计算机的模拟计算

D. 逻辑型语言是在客户机/服务器系统中用于实现负载分散的程序语言

●在过程式程序设计(①)、数据抽象程序设计(②)、面向对象程序设计(③)、泛型(通用)程序设计(④)中，C++ 语言支持 (21) ，C 语言支持 (22) 。

(21) A. ①　　B. ②③　　C. ③④　　D. ①②③④

(22) A. ①　　B. ①③　　C. ②③　　D. ①②③④

●在 UNIX 操作系统中，把输入/输出设备看作 (23) 。

(23) A. 普通文件　　B. 目录文件　　C. 索引文件　　D. 特殊文件

●某系统中有四种互斥资源 R_1、R_2、R_3 和 R_4，可用资源数分别为 3、5、6 和 8。假设在 T_0 时刻有 P_1、P_2、P_3 和 P_4 四个进程，并且这些进程对资源的最大需求量和已分配资源数如下表所示，那么在 T_0 时刻系统中 R_1、R_2、R_3 和 R_4 的剩余资源数分别为 (24) 。如果从 T_0 时刻开始进程按 (25) 顺序逐个调度执行，那么系统状态是安全的。

(24) A. 3、5、6 和 8　　B. 3、4、2 和 2　　C. 0、1、2 和 1　　D. 0、1、0 和 1

(25) A. $P_1 \to P_2 \to P_4 \to P_3$　　B. $P_2 \to P_1 \to P_4 \to P_3$

C. $P_3 \to P_2 \to P_1 \to P_4$　　D. $P_4 \to P_2 \to P_3 \to P_1$

资源 进程	最大需求量				已分配资源数			
	R_1	R_2	R_3	R_4	R_1	R_2	R_3	R_4
P_1	1	2	3	6	1	1	2	4
P_2	1	1	2	2	0	1	2	2
P_3	1	2	1	1	1	1	1	0
P_4	1	1	2	3	1	1	1	1

●若文件系统容许不同用户的文件具有相同的文件名，则操作系统应采用 (26) 来实现。

(26) A. 索引表　　B. 索引文件　　C. 指针　　D. 多级目录

●某虚拟存储系统采用最近最少使用(LRU)页面淘汰算法，假定系统为每个作业分配 3 个页面的主存空间，其中一个页面用来存放程序。现有某作业的部分语句如下：

```
Var A: Array[1..150,1..100] OF integer;
    i,j: integer;
    FOR i:=1 to 150 DO
        FOR j:=1 to 100 DO
            A[i,j]:=0;
```

设每个页面可存放 150 个整数变量，变量 i、j 放在程序页中。初始时，程序及变量 i、j 已在内存，其余两页为空，矩阵 $\boldsymbol{A}$ 按行序存放。在上述程序片段执行过程中，共产生 (27) 次缺页中断。最后留在内存中的是矩阵 $\boldsymbol{A}$ 的最后 (28) 。

(27) A. 50　　B. 100　　C. 150　　D. 300

(28) A. 2 行　　B. 2 列　　C. 3 行　　D. 3 列

●统一过程(UP)的基本特征是“用例驱动、以架构为中心的和受控的迭代式增量开发”。UP 将一个周期的开发过程划分为四个阶段，其中 (29) 的提交结果包含了系统架构。

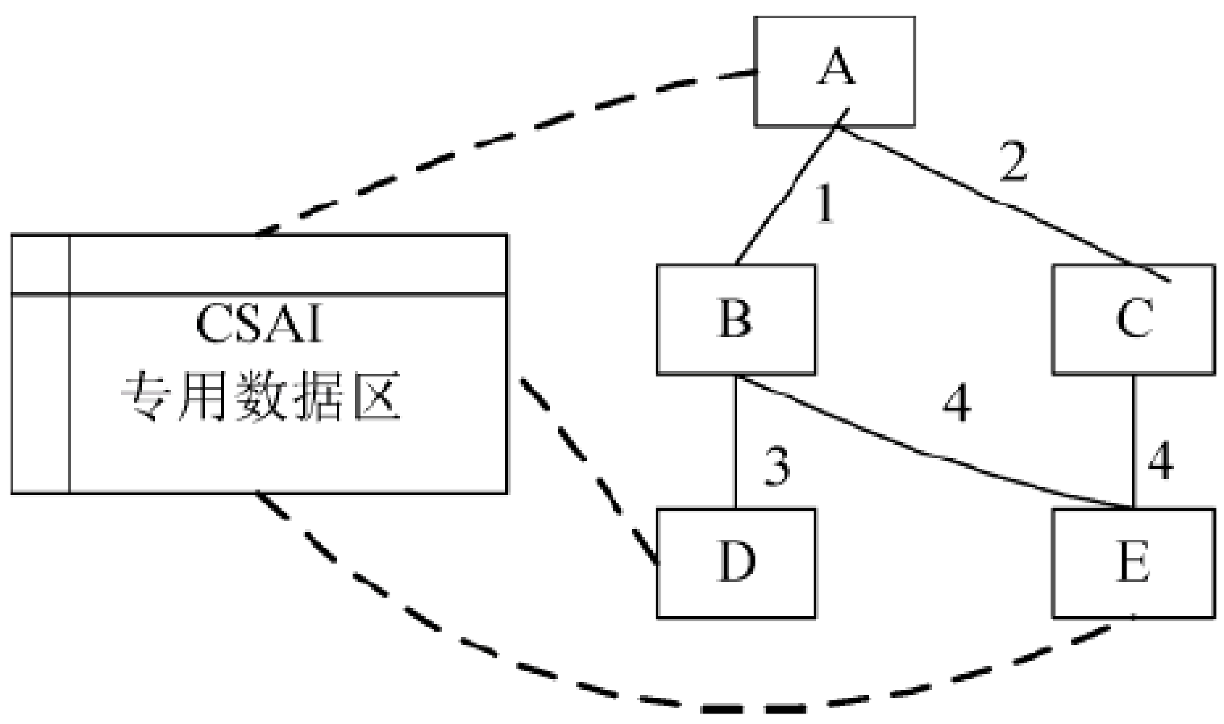
(29) A. 初始阶段　　B. 精化阶段　　C. 构建阶段　　D. 提交阶段

●为验证程序模块 A 是否正确实现了规定的功能，需要进行 (30) ；为验证模块 A 能否与其他模块按照规定方式正确工作，需要进行 (31) 。

(30) A. 单元测试　　B. 集成测试　　C. 确认测试　　D. 系统测试

(31) A. 单元测试　　B. 集成测试　　C. 确认测试　　D. 系统测试

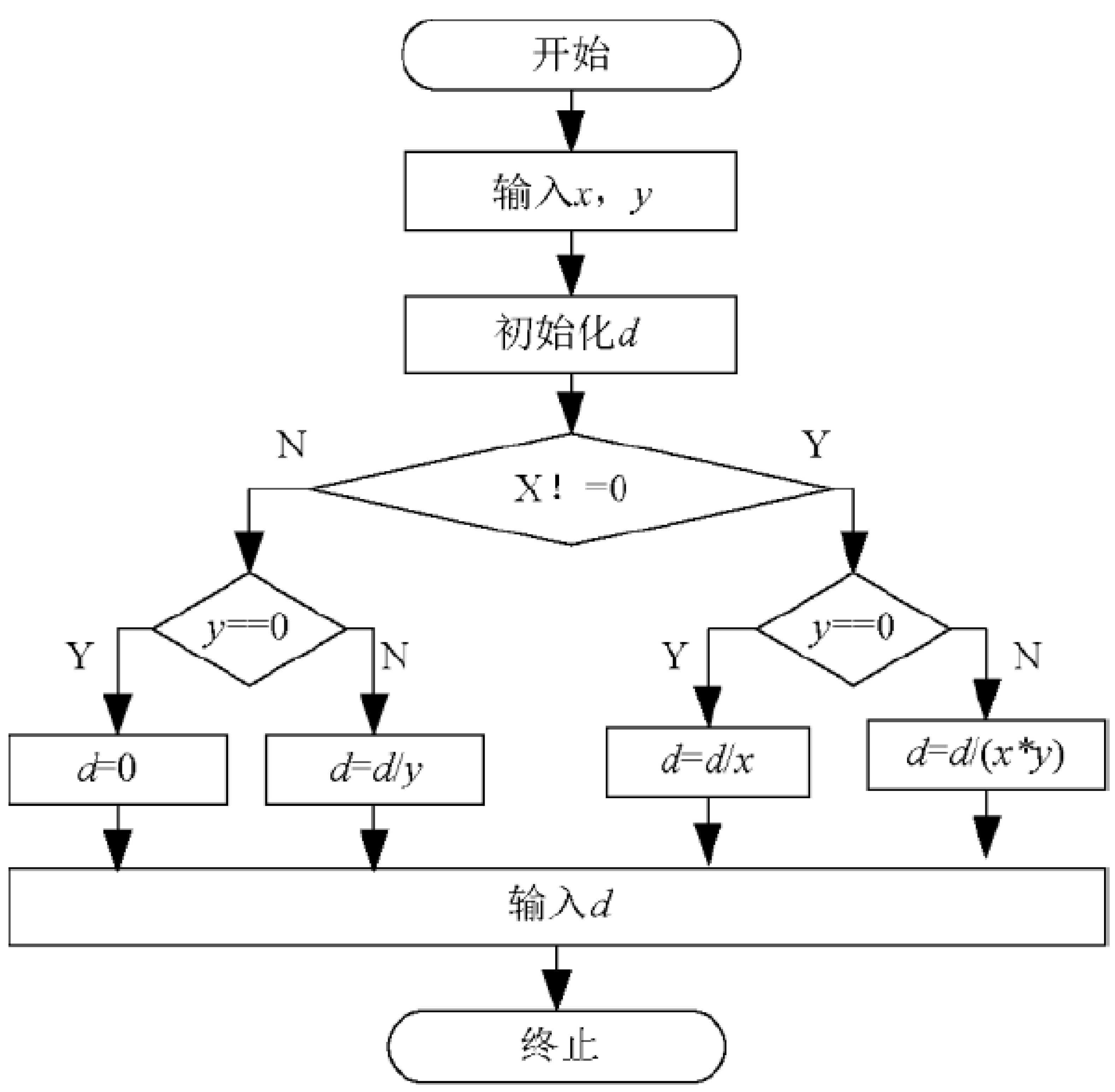
●下图中的程序由 A、B、C、D、E 5 个模块组成，下表中描述了这些模块之间的接口，每一个接口有一个编号。此外，模块 A、D 和 E 都要引用一个专用数据区。那么 A 和 E 之间的耦合关系是 (32) 。



编　号	参　数	返回值
1	数据项	数据项
2	数据项	数据项
3	功能码	无
4	无	列表

(32) A. 公共耦合　　B. 数据耦合　　C. 内容耦合　　D. 无耦合

●阅读下列流程图。



当用判定覆盖法进行测试时，至少需要设计 (33) 个测试用例。

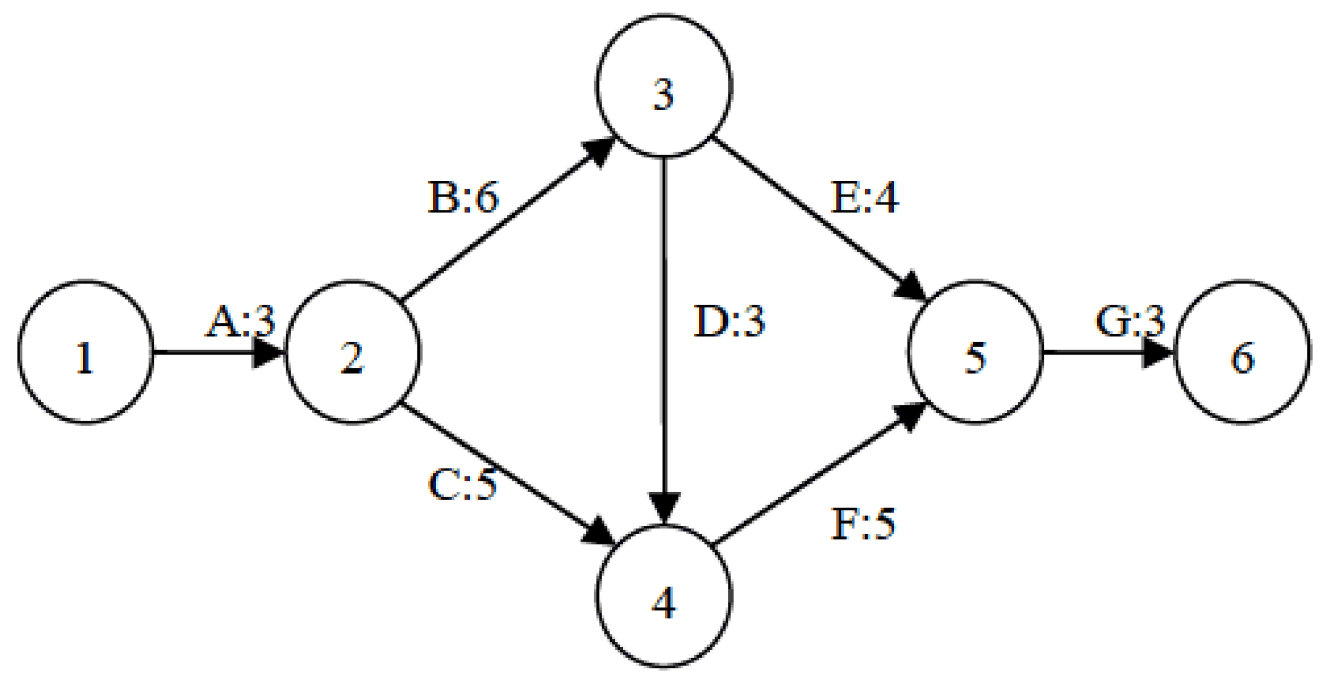
(33) A. 2　　B. 4　　C. 6　　D. 8

●ISO/IEC 9126 软件质量模型中第一层定义了六个质量特性，并为各质量特性定义了相应的质量子特

性，其中易分析子特性属于软件的 (34) 质量特性。

(34) A. 可靠性　B. 效率　C. 可维护性　D. 功能性

●某工程计划图如下图所示，弧上的标记为作业编码及其需要的完成时间(天)，作业 E 最迟应在第 (35) 天开始。



(35) A. 7　B. 9　C. 12　D. 13

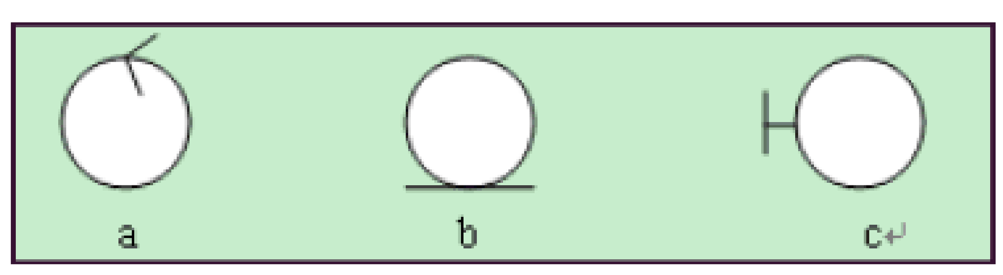
●LOC 是软件规模的一种量度，它表示 (36) 。

(36) A. 软件功能数　B. 源代码行数　C. 每单位成本数　D. 工作量

●采用 UML 进行软件建模过程中，类图是系统的一种静态视图，用 (37) 可明确表示两类事物之间存在的整体/部分形式的关联关系。

(37) A. 依赖关系　B. 聚合关系　C. 泛化关系　D. 实现关系

●在 UML 语言中，下图中的 a、b、c 三种图形符号按照顺序分别表示 (38) 。



(38) A. 边界对象、实体对象、控制对象　B. 实体对象、边界对象、控制对象
C. 控制对象、实体对象、边界对象　D. 边界对象、控制对象、实体对象

●UML 中有多种类型的图，其中， (39) 对系统的使用方式进行分类， (40) 显示了类及其相互关系， (41) 显示人或对象的活动，其方式类似于流程图，通信图显示在某种情况下对象之间发送的消息， (42) 与通信图类似，但强调的是顺序而不是连接。

(39) A. 用例图　B. 顺序图　C. 类图　D. 活动图

(40) A. 用例图　B. 顺序图　C. 类图　D. 活动图

(41) A. 用例图　B. 顺序图　C. 类图　D. 活动图

(42) A. 用例图　B. 顺序图　C. 类图　D. 活动图

●在面向对象的语言中， (43) 。

(43) A. 类的实例化是指对类的实例分配存储空间　B. 每个类都必须创建一个实例
C. 每个类只能创建一个实例　D. 类的实例化是指对类进行初始化

● (44) 设计模式定义了对象间的一种一对多的依赖关系，以便当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动刷新。

(44) A. Adapter(适配器)　B. Iterator(迭代器)
C. Prototype(原型)　D. Observer(观察者)

●面向对象分析需要找出软件需求中客观存在的所有实体对象(概念)，然后归纳、抽象出实体类。 (45)

是寻找实体对象的有效方法之一。

(45) A. 会议调查　　B. 问卷调查　　C. 电话调查　　D. 名词分析

●在进行面向对象设计时，采用设计模式能够__(46)__。

(46) A. 复用相似问题的相同解决方案　　B. 改善代码的平台可移植性
C. 改善代码的可理解性　　D. 增强软件的易安装性

●在采用标准 UML 构建的用例模型(Use Case Model)中，参与者(Actor)与用例(Use Case)是模型中的主要元素，其中参与者与用例之间可以具有__(47)__关系。

(47) A. 包含(Include)　　B. 递归(Recursive)
C. 关联(Association)　　D. 组合(Composite)

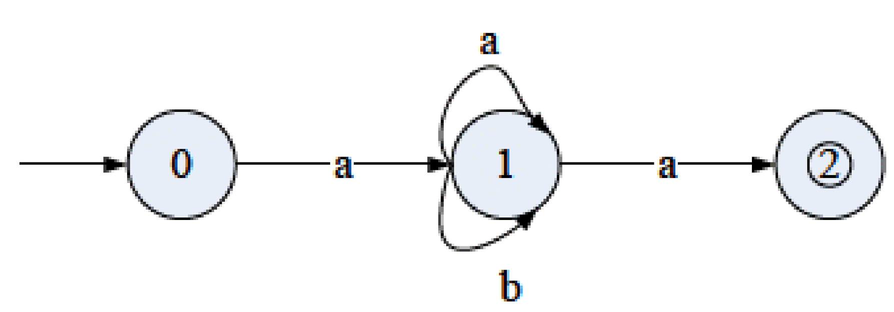
●对于下面的文法 G[S]，__(48)__是其句子(从 S 出发开始推导)。

```
G(S): S→M1(S,M)   M→*P|MP   P→a|b|c|…|x|x|z
```

(48) A. ((a,0))　　B. ((fac,bb),g)　　C. (abc)　　D. (c,(da))

●下图是一有限自动机的状态转换图，该自动机所识别语言的特点是__(49)__，等价的正规式为__(50)__。

(49) A. 由符号 a、b 构成且包含偶数个 a 的串
B. 由符号 a、b 构成且开头和结尾符号都为 a 的串
C. 由符号 a、b 构成的任意串
D. 由符号 a、b 构成且 b 的前后必须为 a 的串



(50) A. (a|b)*(aa)*　　B. a(a|b)*a　　C. (a|b)*　　D. a(ba)*a

●关系 R、S 如下图所示，关系代数表达式 $\pi_{1,5,6}(\sigma_{1>5}(R\times S))$=__(51)__。

关系 R

A	B	C
4	5	6
7	8	9
10	11	12

关系 S

A	B	C
4	7	6
5	12	13
6	10	14

(51) A.

A	B	C
1	12	13
1	10	14

B.

A	B	C
10	7	11
10	7	6

C.

A	B	C
7	12	13
7	10	14

D.

A	B	C
4	7	6
4	7	11

●给定供应关系 SPJ(供应商号，零件号，工程号，数量)，查询某工程至少用了 3 家供应商(包含 3 家)供应的零件的平均数量，并按工程号的降序排列。

```
SELECT 工 程号, __(52)__ FROM SPJ GROUP BY 工程号 __(53)__
ORDER BY 工程号 DESC;
```

(52) A. AVG(数量)At 平均数量　　B. AVG(数量)AS 平均数量

C. 平均数量 At AVG(数量)　　D. 平均数量 AS AVG (数量)

(53) A. HAVING COUNT(DISTINCT(供应商号))>2

B. Where COUNT(供应商号)>2

C. HAVING(DISTINCT(供应商号))>2

D. Where 供应商号 2

●若某个关系的主码为全码，则该主码应包含 (54) 。

(54) A. 单个属性　　B. 两个属性　　C. 多个属性　　D. 全部属性

●关系 R、S 如下图所示，元组演算表达式 $\{t\,|\,(\forall u)(R(t)\wedge S(u)\wedge t[3]>u[1])\}$ 的结果为 (55) 。

关系 R

A	B	C
1	2	3
4	5	6
7	8	9
10	11	12

关系 S

A	B	C
3	7	11
4	5	6
5	10	14
6	10	14

(55) A.

A	B	C
1	2	3
4	5	6

B.

A	B	C
2	7	11
5	5	6

C.

A	B	C
7	8	9
10	11	12

D.

A	B	C
5	9	13
6	10	14

●若事务 T1 对数据 A 已加排他锁，那么其他事务对数据 A (56) 。

(56) A. 加共享锁成功，加排他锁失败　　B. 加排他锁成功，加共享锁失败

C. 加共享锁、加排他锁都成功　　D. 加共享锁、加排他锁都失败

●拓扑排序是指有向图中的所有顶点排成一个线性序列的过程，若在有向图中从顶点 v_i 到 v_j 有一条路径，则在该线性序列中，顶点 v_i 必然在顶点 v_j 之前。因此，若不能得到全部顶点的拓扑排序序列，则说明该有向图一定 (57) 。

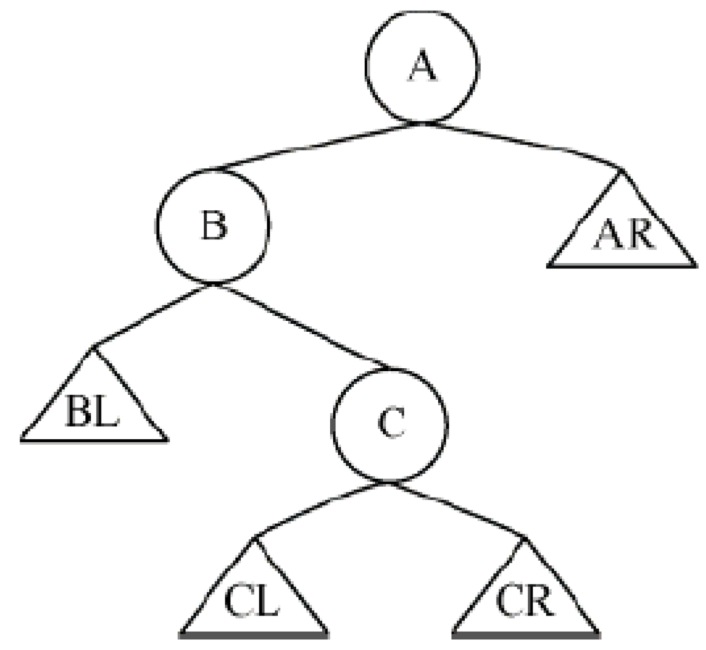
(57) A. 包含回路　　B. 是强连通图　　C. 是完全图　　D. 是有向树

●对于二叉查找树(Binary Search Tree)，若其左子树非空，则左子树上所有结点的值均小于根结点的值；若其右子树非空，则右子树上所有结点的值均大于根结点的值；左、右子树本身就是两棵二叉查找树。因此，对任意一棵二叉查找树进行 (58) 遍历可以得到一个结点元素的递增序列。在具有 n 个结点的二叉查找树上进行查找运算，最坏情况下的算法复杂度为 (59) 。

(58) A. 先序　　B. 中序　　C. 后序　　D. 层序

(59) A. $O(n^2)$　　B. $O(n\log_2 n)$　　C. $O(\log_2 n)$　　D. $O(n)$

●下图所示平衡二叉树(树中任一结点的左右子树高度之差不超过 1)中，结点 A 的右子树 AR 高度为 h，结点 B 的左子树 BL 高度为 h，结点 C 的左子树 CL、右子树 CR 的高度都为 h−1。若在 CR 中插入一个结点并使得 CR 的高度增加 1，则该二叉树 (60) 。



(60) A. 以 B 为根的子二叉树变为不平衡　　B. 以 C 为根的子二叉树变为不平衡
C. 以 A 为根的子二叉树变为不平衡　　D. 仍然是平衡二叉树

●对 n 个元素的数组进行 (61) ，其平均时间复杂度和最坏情况下的时间复杂度都是 $O(n\log_2 n)$。

(61) A. 希尔排序　　B. 快速排序　　C. 堆排序　　D. 选择排序

●对于具有 n 个元素的一个数据序列，若只需得到其中第 k 个元素之前的部分排序，最好采用 (62) ，使用分治(divide and conquer)策略的是 (63) 算法。

(62) A. 希尔排序　　B. 直接插入排序　　C. 快速排序　　D. 堆排序

(63) A. 冒泡排序　　B. 插入排序　　C. 快速排序　　D. 堆排序

●迪杰斯特拉(Dijkstra)算法按照路径长度递增的方式求解单源点最短路径问题，该算法运用了 (64) 算法策略。

(64) A. 贪心　　B. 分而治之　　C. 动态规划　　D. 试探+回溯

●设某算法的计算时间可用递推关系式 $T(n)=2T(n/2)+n$ 表示，则该算法的时间复杂度为 (65) 。

(65) A. $O(\lg n)$　　B. $O(n\lg n)$　　C. $O(n)$　　D. $O(n^2)$

●关于 ARP 表，以下描述中正确的是 (66) 。

(66) A. 提供常用目标地址的快捷方式来减少网络流量
B. 用于建立 IP 地址到 MAC 地址的映射
C. 用于在各个子网之间进行路由选择
D. 用于进行应用层信息的转换

●在 Windows 操作系统中，采用 (67) 命令来测试到达目标所经过的路由器数目及 IP 地址。

(67) A. ping　　B. tracert　　C. arp　　D. nslookup

●在 FTP 中，控制连接是由 (68) 主动建立的。

(68) A. 服务器端　　B. 客户端　　C. 操作系统　　D. 服务提供商

●在进行金融业务系统的网络设计时，应该优先考虑 (69) 原则。在进行企业网络的需求分析时，应该首先进行 (70) 。

(69) A. 先进性　　B. 开放性　　C. 经济性　　D. 高可用性

(70) A. 企业应用分析　　B. 网络流量分析
C. 外部通信环境调研　　D. 数据流向图分析

● (71) analysis emphasizes the drawing of pictorial system models to document and validate both existing and/or proposed systems. Ultimately, the system models become the (72) for designing and constructing an improved system. (73) is such a technique. The emphasis in this technique is process-centered. Systems analysts draw a series of process models called (74) . (75) is another such technique that integrates data and process concerns into constructs called objects.

(71) A. Prototyping　　B. Accelerated　　C. Model-driven　　D. Iterative

(72)A. image　　B. picture　　C. layout　　D. blueprint

(73) A. Structured analysis　　B. Information Engineering

C. Discovery Prototyping　　D. Object-Oriented analysis

(74) A. PERT　　B. DFD　　C. ERD　　D. UML

(75) A. Structured analysis　　B. Information Engineering

C. Discovery Prototyping　　D. Object-Oriented analysis

下午科目

说明：试题五、试题六选做一题。

试题一

阅读以下说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某高校欲开发一个成绩管理系统，记录并管理所有选修课程的学生的平时成绩和考试成绩，其主要功能描述如下。

(1) 每门课程都由 3～6 个单元构成，每个单元结束后会进行一次测试，其成绩作为这门课程的平时成绩。课程结束后进行期末考试，其成绩作为这门课程的考试成绩。

(2) 学生的平时成绩和考试成绩均由每门课程的主讲教师上传给成绩管理系统。

(3) 在记录学生成绩之前，系统需要验证这些成绩是否有效。首先，根据学生信息文件来确认该学生是否选修这门课程，若没有，那么这些成绩是无效的；如果他的确选修了这门课程，再根据课程信息文件和课程单元信息文件来验证平时成绩是否与这门课程所包含的单元相对应，如果是，那么这些成绩是有效的，否则无效。

(4) 对于有效成绩，系统将其保存在课程成绩文件中。对于无效成绩，系统会单独将其保存在无效成绩文件中，并将详细情况提交给教务处。在教务处没有给出具体处理意见之前，系统不会处理这些成绩。

(5) 若一门课程的所有有效的平时成绩和考试成绩都已经被系统记录，系统会发送课程完成通知给教务处，告知该门课程的成绩已经齐全。教务处根据需要，请求系统生成相应的成绩列表，用来提交考试委员会审查。

(6) 在生成成绩列表之前，系统会生成一份成绩报告给主讲教师，以便核对是否存在错误。主讲教师须将核对之后的成绩报告返还系统。

(7) 根据主讲教师核对后的成绩报告，系统生成相应的成绩列表，递交考试委员会进行审查。考试委员会在审查之后，上交一份成绩审查结果给系统。对于所有通过审查的成绩，系统将会生成最终的成绩单，并通知每个选课学生。

现采用结构化方法对这个系统进行分析与设计，得到如图 1 所示的顶层数据流图和图 2 所示的 0 层数据流图。

【问题 1】

使用说明中的词语，给出图 1 中的外部实体 E1～E4 的名称。

【问题 2】

使用说明中的词语，给出图 2 中的数据存储 D1～D5 的名称。

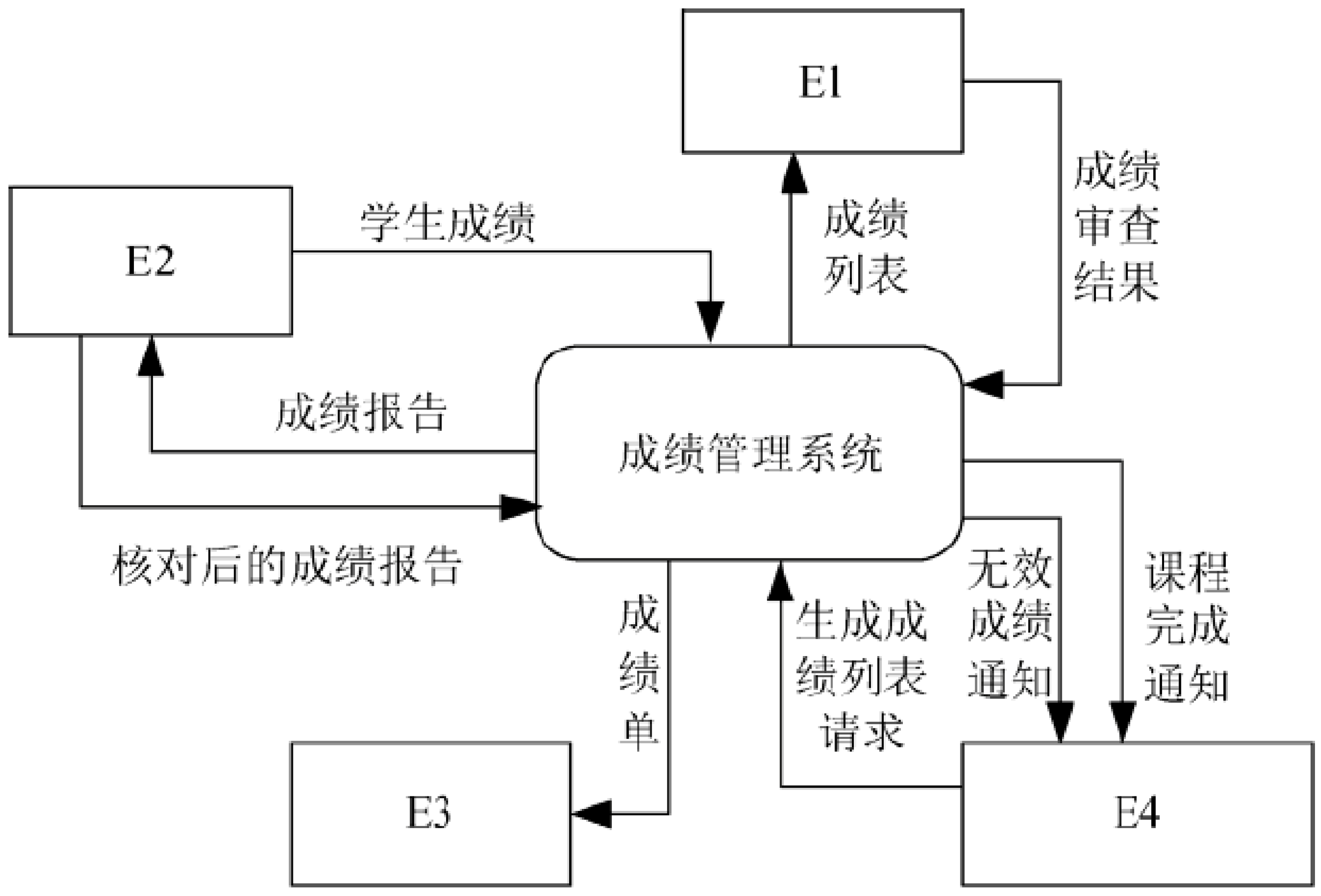


图 1　顶层数据流图

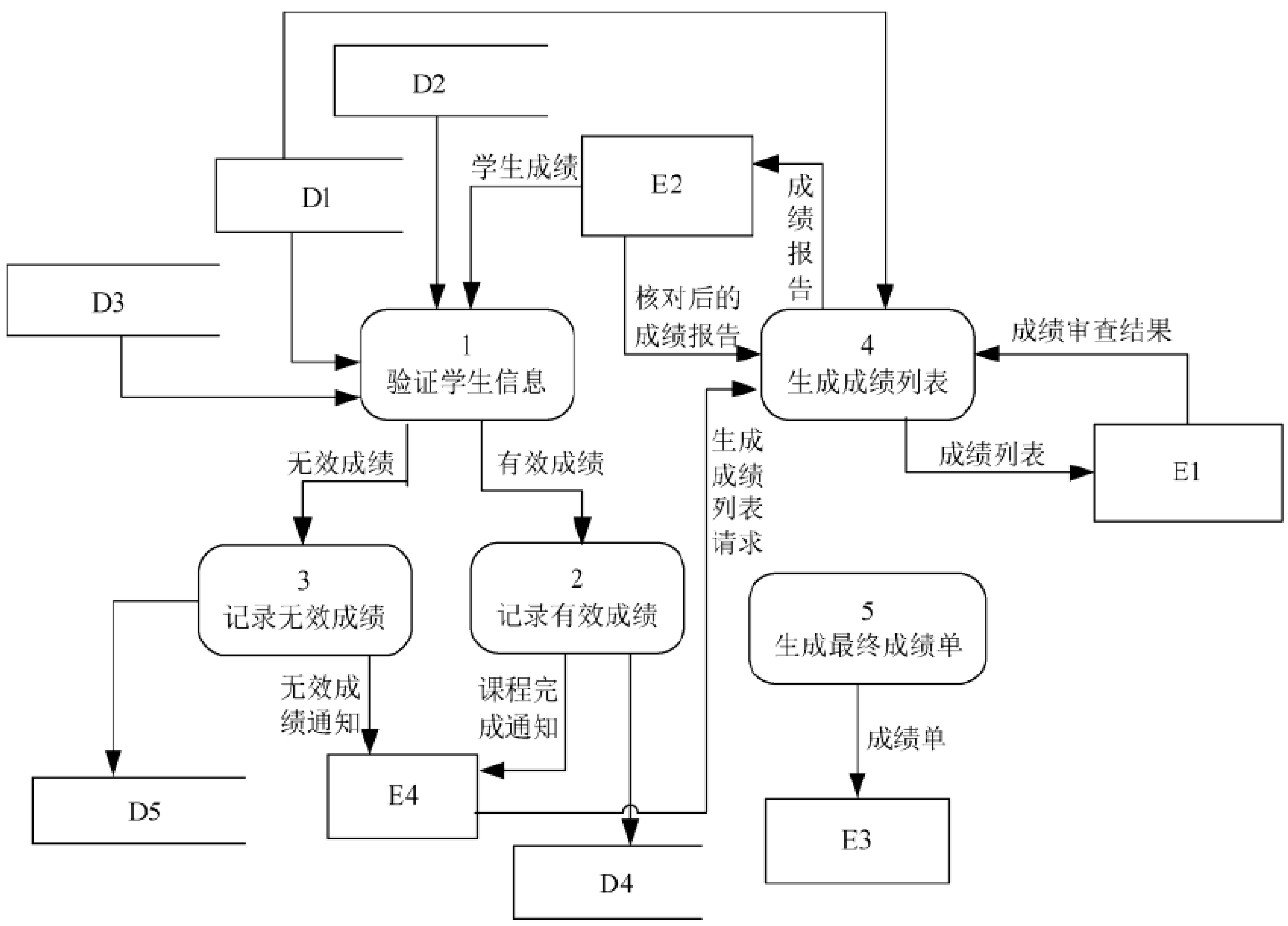


图 2　0 层数据流图

【问题 3】

数据流图 2 缺少了三条数据流，根据说明及数据流图 1 提供的信息，分别指出这三条数据流的起点和终点。

起 点	终 点

【问题 4】

数据流图是在系统分析与总体设计阶段宏观地描述系统功能需求的重要图形化工具，程序流程图也是

软件开发过程中比较常用的图形化工具。简要说明程序流程图的适用场合与作用。

试题二

阅读下列说明，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某医院的门诊管理系统实现了为患者提供挂号、处方药品收费的功能，具体的需求及设计如下。

(1) 医院医师具有编号、姓名、科室、职称、出诊类型和出诊费用，其中出诊类型分为专家门诊和普通门诊，与医师职称无关；各个医师可以具有不同的出诊费用，与职称和出诊类型无关。

(2) 患者首先在门诊挂号处挂号，选择科室和医师，根据选择的医师缴纳挂号费(医师出诊费)。收银员为患者生成挂号单，如表 1 所示，其中，就诊类型为医师的出诊类型。

表 1　XX 医院门诊挂号单

收银员：13011　　　　时间：2012年2月1日 08:58

就诊号	姓　名	科　室	医　师	就诊类型	挂号费/元
20120205015	叶萌	内科	杨玉明	专家门诊	5

(3) 患者在医师处就诊后，凭借挂号单和医师手写处方到门诊药房交费买药。收银员根据就诊号和医师处方中开列的药品信息，查询药品库(如表 2 所示)，并生成门诊处方单(如表 3 所示)。

表 2　药品库

药品编码	药品名称	类　型	库　存	货架编号	单　位	规　格	单价/元
12007	牛蒡子	中药	51590	B1410	G	炒	0.0340
11090	百部	中药	36950	B1523	G	片	0.0313

表 3　XX 医院门诊处方单

时间：2012 年 2 月 1 日 10:31

就诊号	20120205015	病人名称	叶萌	医师姓名	杨玉明
金额总计	0.65	项目总计	2	收银员	21081
药品编码	药品名称	数量	单位	单价	金额/元
12007	牛蒡子	10	G	0.0340	0.34
11090	百部	10	G	0.0313	0.31

(4) 由于药品价格会发生变化，因此，门诊管理系统必须记录处方单上药品的单价。

根据需求阶段收集的信息，设计的实体联系图如图 1 所示，关系模式(不完整)。

① 实体联系图。

② 关系模式。

挂号单(就诊号,病患姓名,医师编号,时间, __(5)__)

收银员(编号,姓名,级别)

医师(编号,姓名,科室,职称,出诊类型,出诊费用)

门诊处方(__(6)__ ,收银员,时间)

处方明细(就诊号, (7))

药品库(药品编码,药品名称, (8))

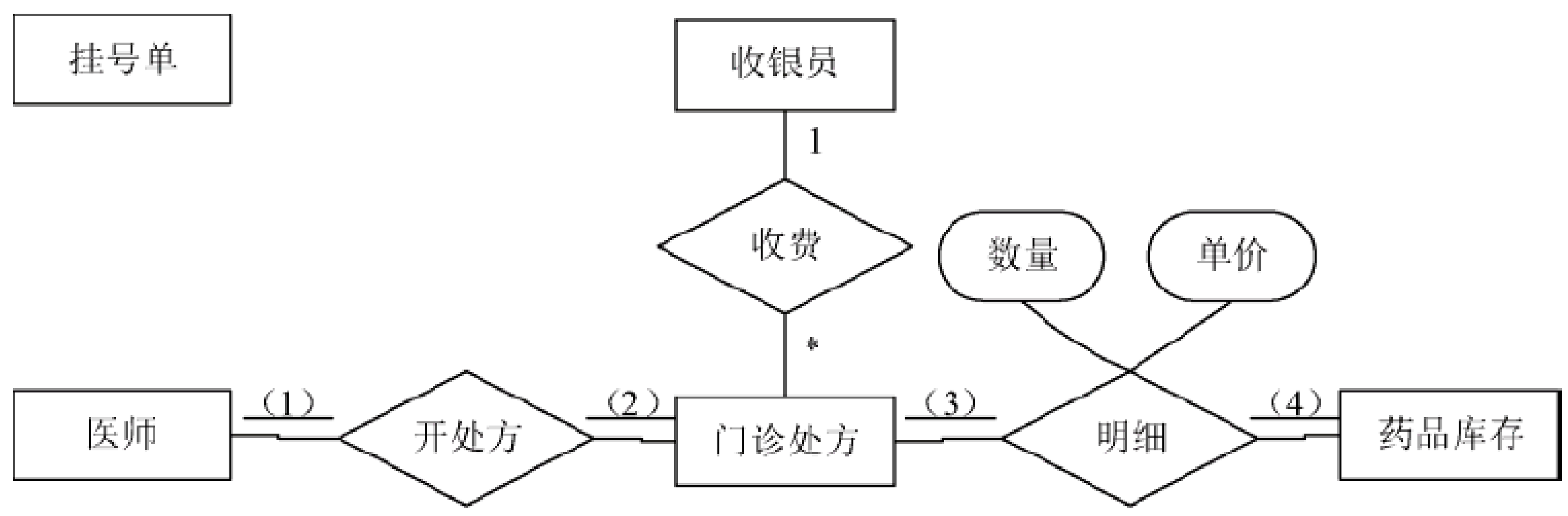


图 1　实体联系图

【问题 1】

根据问题描述，填写图 1 中(1)～(4)处联系的类型。

【问题 2】

图 1 中还缺少几个联系？指出每个联系两端的实体名，格式如下。

实体 1：实体 2

例如，收银员与门诊处方之间存在联系，表示为：收银员：门诊处方或门诊处方：收银员

【问题 3】

根据实体联系图 1，填写挂号单、门诊处方、处方明细和药品库关系模式中的空(5)～(8)处，并指出挂号单、门诊处方和处方明细关系模式的主键。

试题三

阅读下列说明和图，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

已知某唱片播放器不仅可以播放唱片，而且可以连接电脑并把电脑中的歌曲刻录到唱片上(同步歌曲)。连接电脑的过程中还可自动完成充电。

关于唱片，还有以下描述信息。

(1) 每首歌曲的描述信息包括歌曲的名字、谱写这首歌曲的艺术家以及演奏这首歌曲的艺术家。只有两首歌曲的这三部分信息完全相同时，才认为它们是同一首歌曲。艺术家可能是一名歌手或一支由 2 名或 2 名以上的歌手所组成的乐队。一名歌手可以不属于任何乐队，也可以属于一个或多个乐队。

(2) 每张唱片由多条音轨构成；一条音轨中只包含一首歌曲或为空，一首歌曲可分布在多条音轨上；同一首歌曲在一张唱片中最多只能出现一次。

(3) 每条音轨都有一个开始位置和持续时间。一张唱片上音轨的次序是非常重要的，因此对于任意一条音轨，播放器需要准确地知道，它的下一条音轨和上一条音轨是什么(如果存在的话)。

根据上述描述，采用面向对象方法对其进行分析与设计，得到了如表 1 所示的类列表、如图 1 所示的初始类图以及如图 2 所示的描述播放器行为的 UML 状态图。

表 1　类列表

类　名	说　明
Artist	艺术家
Song	歌曲
Band	乐队
Musician	歌手
Track	音轨
Album	唱片

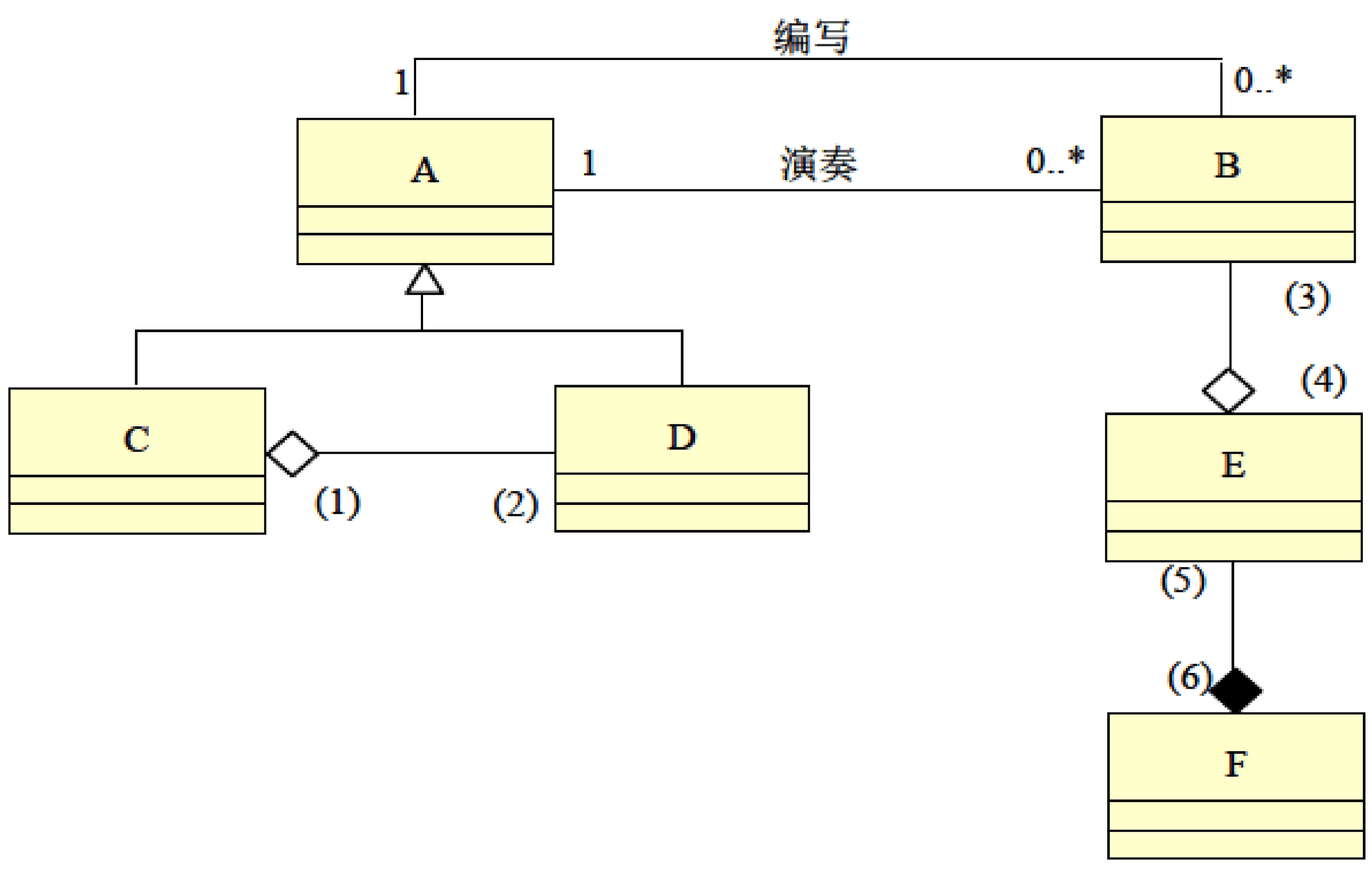


图 1　初始类图

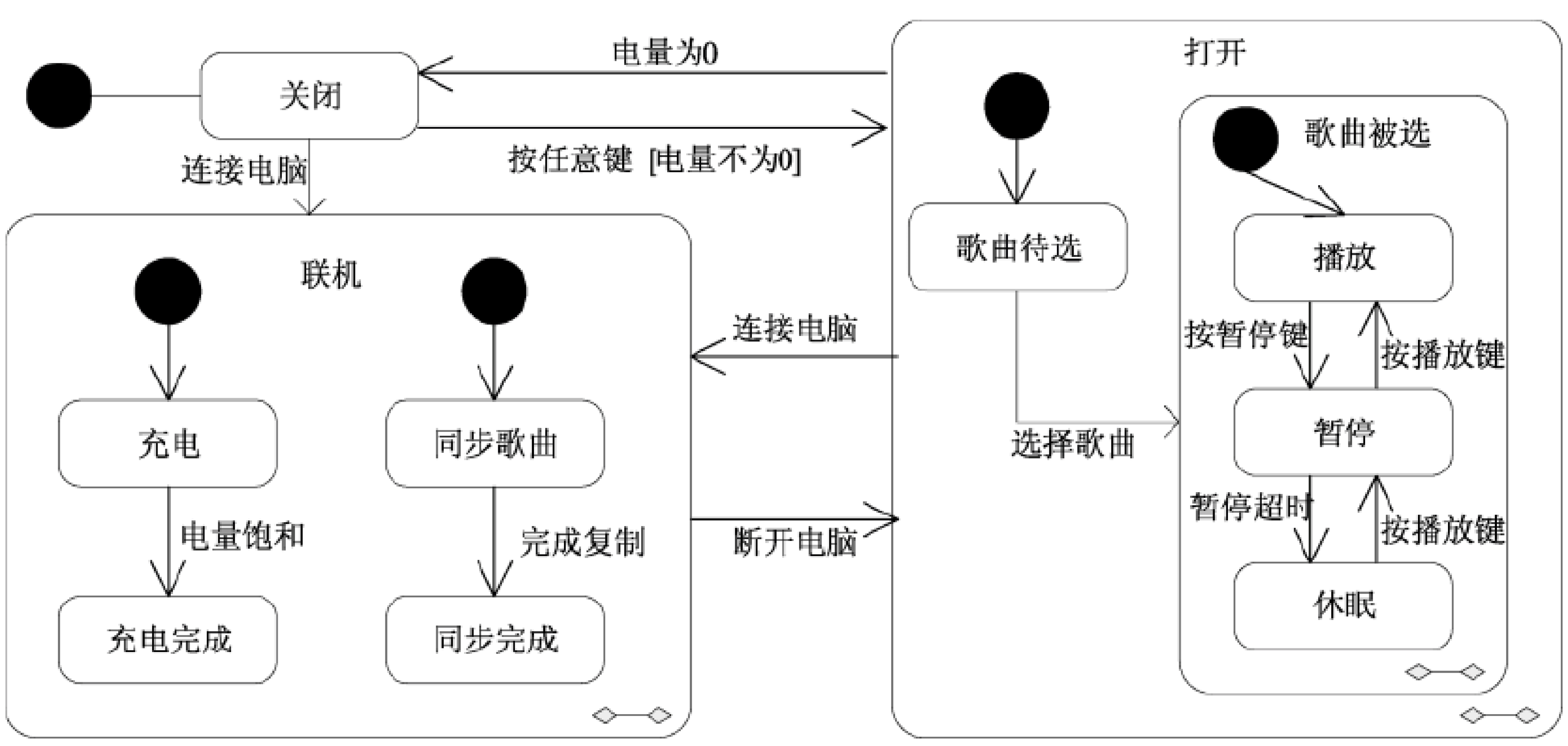


图 2　播放器行为 UML 状态图

【问题 1】

根据说明中的描述，使用表 1 给出的类的名称，给出图 1 中的 A～F 所对应的类。

【问题 2】

根据说明中的描述，给出图 2 中(1)～(6)处的多重度。

【问题 3】

图 1 中缺少了一条关联，请指出这条关联两端所对应的类以及每一端的多重度。

类	多重度

【问题 4】

根据图 2 所示的播放器行为 UML 状态图，给出从“关闭”状态到“播放”状态所经过的最短事件序列(假设电池一开始就是有电的)。

试题四

阅读以下说明和 C 语言函数，将应填入 *(n)* 处的字句写在答题纸的对应栏内。

【说明】

在一个分布网络中，资源(石油、天然气、电力等)可从生产地送往其他地方。在传输过程中，资源会有损耗。例如，天然气的气压会减少，电压会降低。我们将需要输送的资源信息称为信号。在信号从信源地送往消耗地的过程中，仅能容忍一定范围的信号衰减，称为容忍值。分布网络可表示为一个树型结构，如图 1 所示。信号源是树根，树中的每个结点(除了根)表示一个可以放置放大器的子结点，其中某些结点同时也是信号消耗点，信号从一个结点流向其子结点。

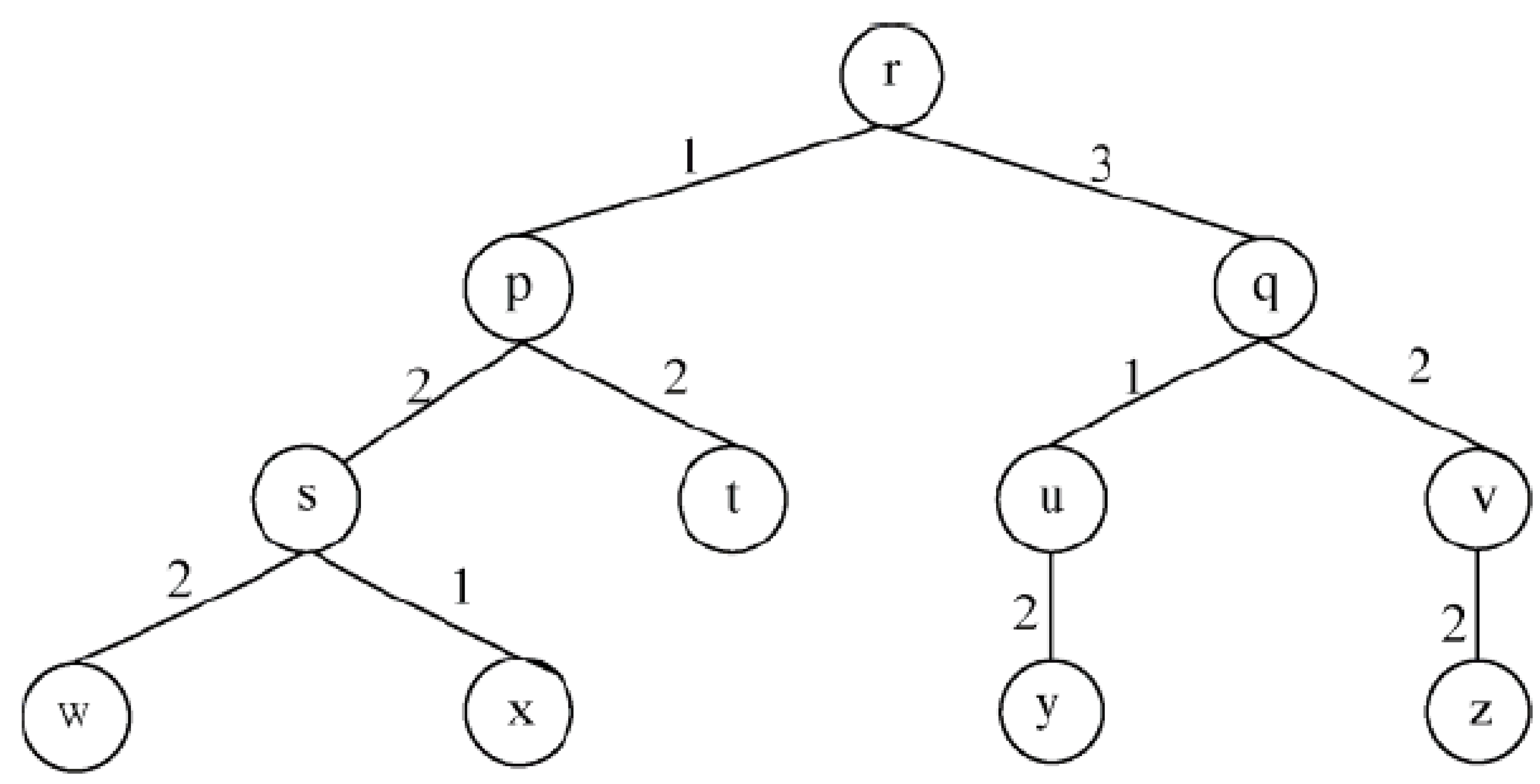


图 1　分布网络的树型结构

每个结点有一个 d 值，表示从其父结点到该结点的信号衰减量。例如，在图 1 中，结点 w、p、q 的 d 值分别为 2、1、3，树根结点表示信号源，其 d 值为 0。

每个结点有一个 M 值，表示从该结点出发到其所有叶子的信号衰减量的最大值。显然，叶子结点的 M 值为 0。对于非叶子 j，$M(j)=\max\{M(k) + d(k) \mid k 是 j 孩子结点\}$，在此公式中，要计算结点的 M 值，必须先算出其所有子结点的 M 值。

在计算 M 值的过程中，对于某个结点 i，其有一个子结点 k 满足 d(k)+M(k)大于容忍值，则应在 k 处放置放大器，否则，从结点 i 到某叶子结点的信号衰减量会超过容忍值，使得到达该叶子结点时信号不可用，而在结点 i 处放置放大器并不能解决到达叶子结点的信号衰减问题。例如，在图 1 中，从结点 p 到其所有叶子结点的最大衰减值为 4。若容忍值为 3，则必须在 s 处放置信号放大器，这样可使得结点 p 的 M 值为 2。同样，需要在结点 q、v 处放置信号放大器，如图 2 中的阴影结点所示。若在某结点放置了信号放大器，则从该结点输出的信号源输出的信号等价。

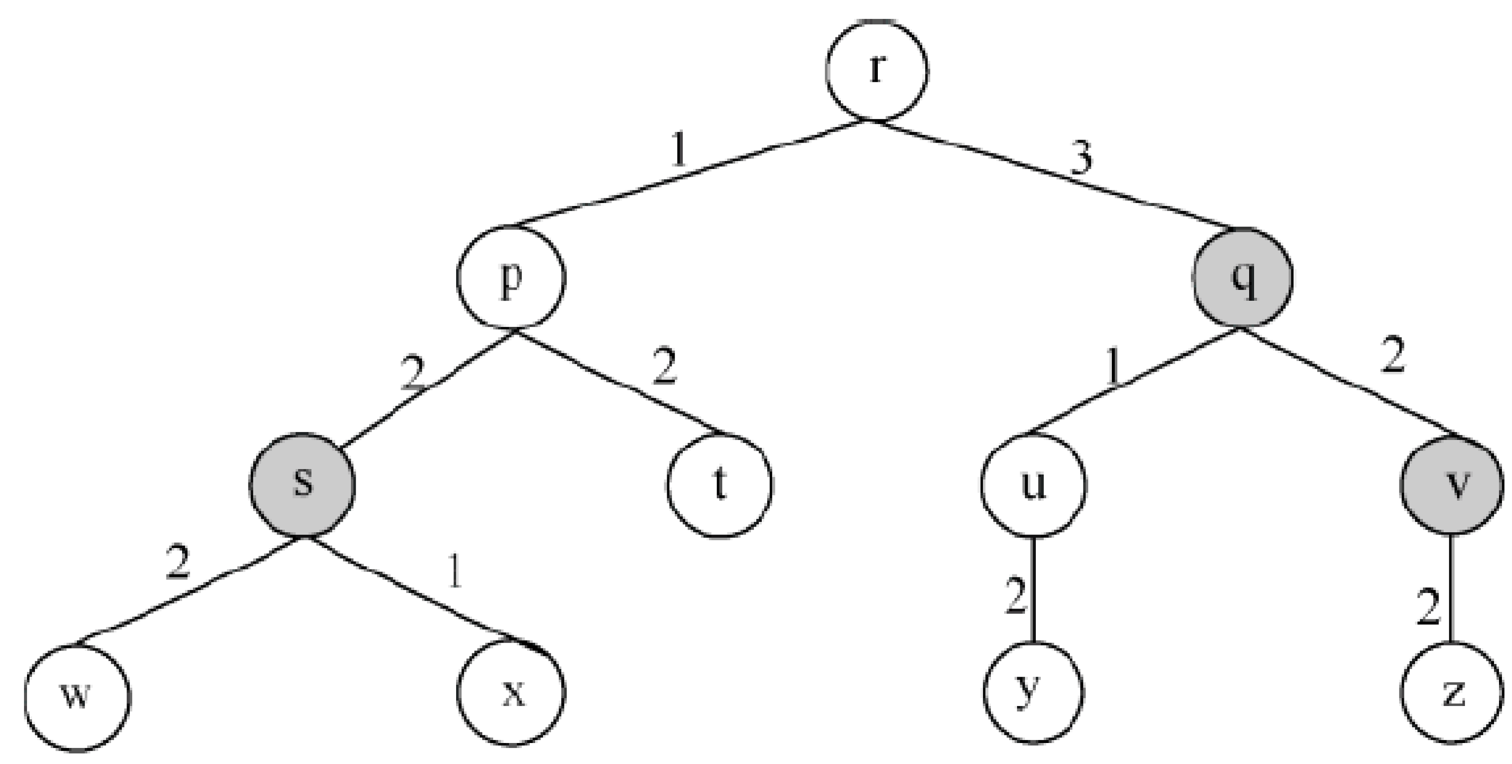


图 2　分布网络中的放大器

函数 placeBoosters(TreeNode *root)的功能是：对于给定树型分布网络中各个结点，计算其信号衰减量的最大值，并确定应在树中的哪些结点放置信号放大器。

全局变量 Tolerance 保存信号衰减容忍值。树的结点类型定义如下：

```
typedef struct TreeNode {
    int id; /*当前结点的识别号*/
    int ChildNum; /*当前结点的子结点数目*/
    int d; /*父结点到当前结点的信号衰减值*/
    struct TreeNode **childptr; /* 向量，存放当前结点到其所有子结点的指针*/
    int M; /*当前结点到其所有子结点的信号衰减值中的最大值*/
    bool boost; /*是否在当前结点放置信号放大器的标志*/
}TreeNode;
```

【C 语言函数】

```
void placeBooster(TreeNode *root)
{   /*计算 root 所指结点处的衰减量，如果衰减量超出容忍值，则放置放大器*/
    TreeNode *p;
    int i,degradation;
    if(__(1)__){
        degradation=0;root->M=0;
        i=0;
        if(i>=root->ChildNum)
            return;
        p=__(2)__;
        for(;i<root->ChildNum && p;i++,p=__(3)__){
            p->M=0;
__(4)__;
if(p->d+p->M>Tolerance){/*在 p 所指结点中放置信号放大器*/
    p->boost=true;
    p->M=0;
}
if(p->d+p->M>degradation)
    degradation=p->d+p->M;
}
        Root->M=__(5)__;
    }
}
```

试题五

阅读下列说明和 C++代码，将应填入__(n)__处的字句写在答题纸的对应栏内。

【说明】

已知某企业的采购审批是分级进行的，即根据采购金额的不同由不同层次的主管人员来审批，主任可以审批 5 万元以下(不包括 5 万元)的采购单，副董事长可以审批 5 万元至 10 万元(不包括 10 万元)的采购单，董事长可以审批 10 万元至 50 万元(不包括 50 万元)的采购单，50 万元及以上的采购单就需要开会讨论决定。

采用责任链设计模式(Chain of Responsibility)对上述过程进行设计后得到的类图如图 1 所示。

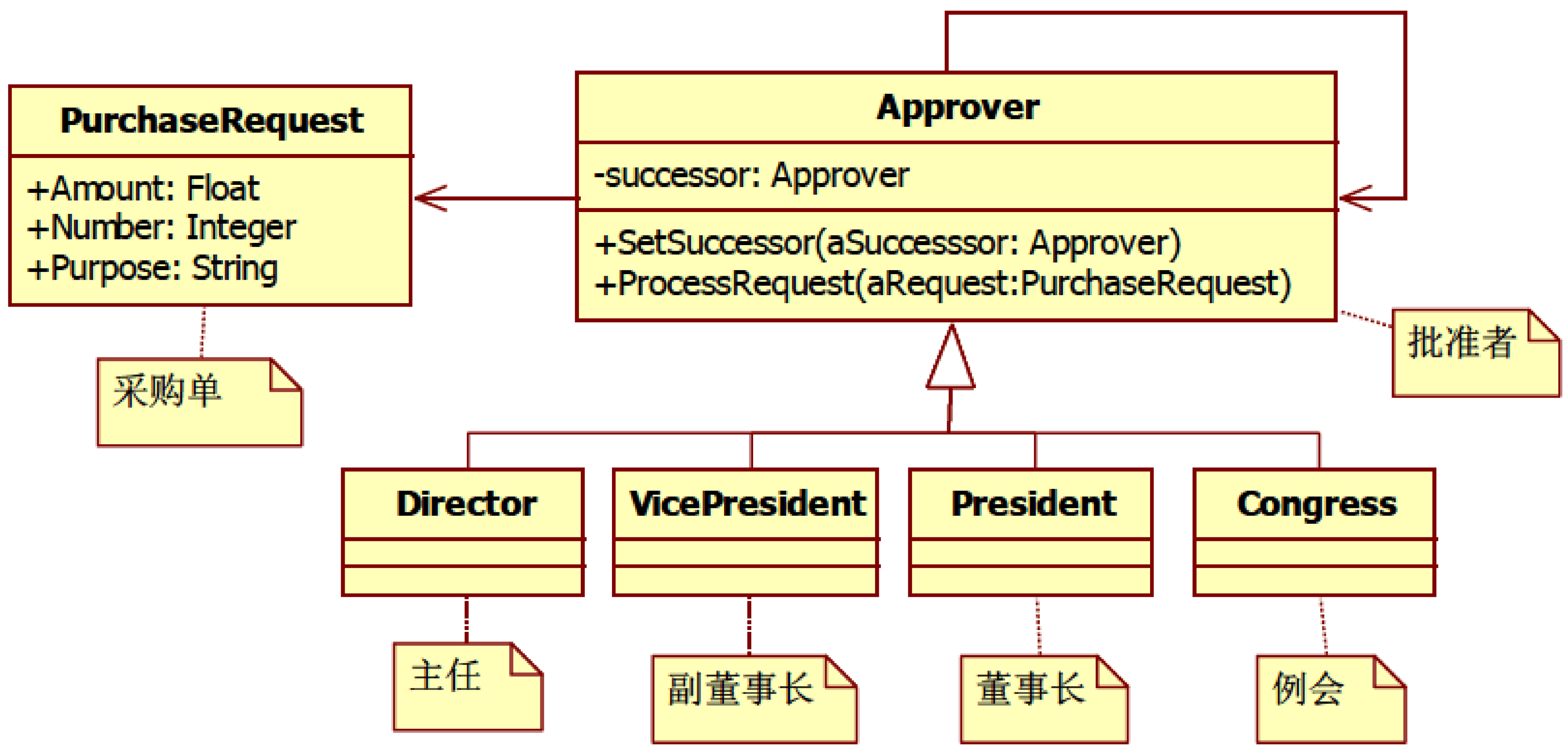


图 1　设计类图

【C++代码】

```
#include <string>
#include <iostream>
using namespace std;
class PurchaseRequest {
    public:
        double Amount;       // 一个采购的金额
        int Number;              // 采购单编号
        string Purpose;      // 采购目的
};
class Approver {             // 审批者类
    public:
        Approver(){ successor = NULL;   }
        virtual void ProcessRequest(PurchaseRequest aRequest){
            if (successor != NULL){  successor-> (1) ;  }
    }
    void SetSuccessor(Approver *aSuccessor){ successor = aSuccessor; }
private:
     (2)  successor;
};
class Congress : public Approver {
public:
    void ProcessRequest(PurchaseRequest aRequest){
        if(aRequest.Amount >= 500000){  /* 决定是否审批的代码省略 */    }
        else  (3)  ProcessRequest(aRequest);
    }
};
class Director : public Approver {
```

```
public:
    void ProcessRequest(PurchaseRequest aRequest){  /* 此处代码省略 */   }
};
class President : public Approver {
public:
    void ProcessRequest(PurchaseRequest aRequest){  /* 此处代码省略 */  }
};
class VicePresident : public Approver {
public:
    void ProcessRequest(PurchaseRequest aRequest){  /* 此处代码省略 */   }
};
void main(){
   Congress Meeting;  VicePresident Sam;  Director Larry ;  President Tammy;

    // 构造责任链
    Meeting.SetSuccessor(NULL);    Sam.SetSuccessor( (4) );
    Tammy.SetSuccessor( (5) );Larry.SetSuccessor( (6) );

    PurchaseRequest aRequest;   // 构造一采购审批请求
    cin >> aRequest.Amount;   // 输入采购请求的金额
     (7) .ProcessRequest(aRequest);     // 开始审批
    return ;
}
```

试题六

阅读下列说明和 Java 代码，将应填入 (*n*) 处的字句写在答题纸的对应栏内。

【说明】

已知某企业的采购审批是分级进行的，即根据采购金额的不同由不同层次的主管人员来审批，主任可以审批 5 万元以下(不包括 5 万元)的采购单，副董事长可以审批 5 万元至 10 万元(不包括 10 万元)的采购单，董事长可以审批 10 万元至 50 万元(不包括 50 万元)的采购单，50 万元及以上的采购单就需要开会讨论决定。

采用责任链设计模式(Chain of Responsibility)对上述过程进行设计后得到的类图如图 1 所示。

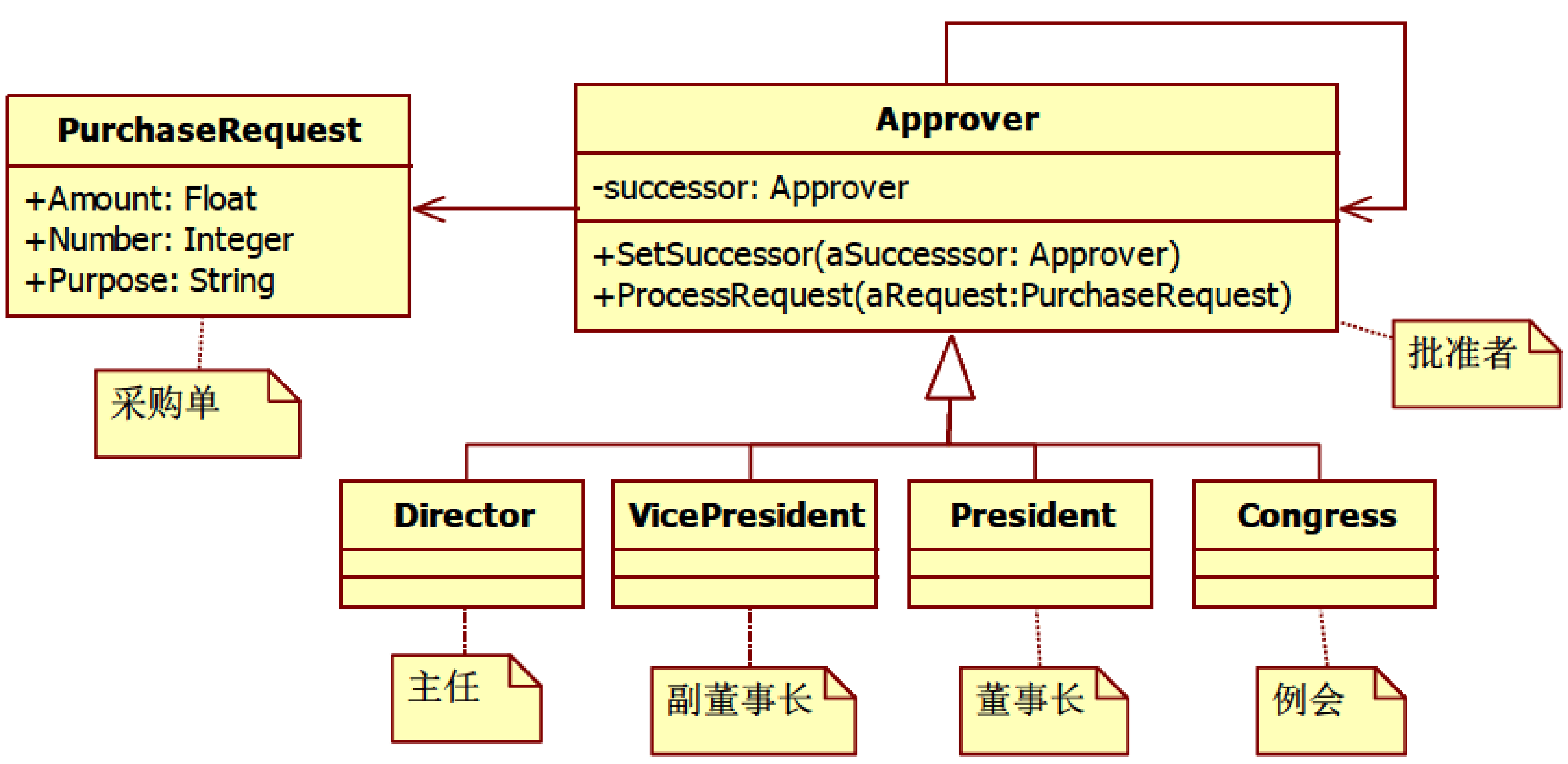


图 1　设计类图

【Java 代码】

```
class PurchaseRequest {
public  double Amount;   // 一个采购的金额
public  int Number;     // 采购单编号
public  String Purpose;   // 采购目的
};
class Approver {       // 审批者类
public   Approver(){    successor = null;   }
public  void ProcessRequest(PurchaseRequest aRequest){
        if (successor != null){ successor. (1) ;    }
    }
public  void SetSuccessor(Approver aSuccesssor){ successor = aSuccesssor; }
private  (2)  successor;
};

class Congress extends Approver {
public  void ProcessRequest(PurchaseRequest aRequest){
        if(aRequest.Amount >= 500000){ /* 决定是否审批的代码省略 */    }
        else  (3) .ProcessRequest(aRequest);
    }
};
class Director extends Approver {
public  void ProcessRequest(PurchaseRequest aRequest){ /* 此处代码省略 */
     }
};
class President extends Approver {
public  void ProcessRequest(PurchaseRequest aRequest){ /* 此处代码省略 */ }
};
class VicePresident extends Approver {
public  void ProcessRequest(PurchaseRequest aRequest){ /* 此处代码省略 */ }
};

public class rs {
    public static void main(String[] args) throws IOException {
        Congress Meeting = new Congress();
        VicePresident Sam = new VicePresident();
        Director Larry = new Director();
        President Tammy = new President();
        // 构造责任链
        Meeting.SetSuccessor(null);     Sam.SetSuccessor( (4) );
        Tammy.SetSuccessor( (5) );      Larry.SetSuccessor( (6) );
// 构造一采购审批请求
        PurchaseRequest aRequest = new PurchaseRequest();
        BufferedReader br =
new BufferedReader(new InputStreamReader(System.in));
        aRequest.Amount = Double.parseDouble(br.readLine());

 (7) .ProcessRequest(aRequest);     // 开始审批
```

```
            return ;
        }
    }
```

13.1.2　模拟试卷二

上午科目

●两个同符号的数相加或异符号的数相减，所得结果的符号位 SF 和进位标志 CF 进行 (1) 运算为 1 时，表示运算的结果产生溢出。

(1) A. 与　　B. 或　　C. 与非　　D. 异或

●在 CPU 与主存之间设置高速缓冲存储器(Cache)，其目的是 (2) 。

(2) A. 扩大主存的存储容量　　B. 提高 CPU 对主存的访问效率
C. 既扩大主存容量又提高存取速度　　D. 提高外存储器的速度

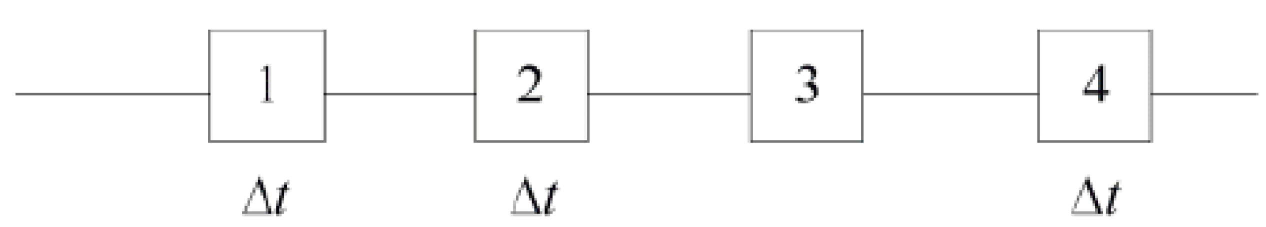
●若内存地址区间为 4000H～43FFH，每个存储单元可存储 16 位二进制数，该内存区域用 4 片存储器芯片构成，则构成该内存所用的存储器芯片的容量是 (3) 。

(3) A. 512×16bit　　B. 256×8bit
C. 256×16bit　　D. 1024×8bit

●在指令系统的各种寻址方式中，获取操作数最快的方式是 (4) 。若操作数的地址包含在指令中，则属于 (5) 方式。

(4) A. 直接寻址　　B. 立即寻址　　C. 寄存器寻址　　D. 间接寻址
(5) A. 直接寻址　　B. 立即寻址　　C. 寄存器寻址　　D. 间接寻址

●指令流水线将一条指令的执行过程分为四步，其中第 1、2 和 4 步的经过时间为Δt，如下图所示。若该流水线顺序执行 50 条指令共用 153Δt，并且不考虑相关问题，则该流水线的第 3 步的时间为 (6) Δt。



(6) A. 2　　B. 3　　C. 4　　D. 5

●驻留在多个网络设备上的程序在短时间内同时产生大量的请求消息冲击某 Web 服务器，导致该服务器不堪重负，无法正常响应其他合法用户的请求，这属于 (7) 。

(7) A. 网上冲浪　　B. 中间人攻击　　C. DDoS 攻击　　D. MAC 攻击

●设有两个子网 202.118.133.0/24 和 202.118.130.0/24，如果进行路由汇聚，得到的网络地址是 (8) 。

(8) A. 202.118.128.0/21　　B. 202.118.128.0/22
C. 202.118.130.0/22　　D. 202.118.132.0/20

●某校园网用户无法访问外部站点 210.102.58.74，管理人员在 Windows 操作系统下可以使用 (9) 判断故障发生在校园网内还是校园网外。

(9) A. ping 210.102.58.74　　B. tracert 210.102.58.74
C. netstat 210.102.58.74　　D. arp 210.102.58.74

● (10) 不属于知识产权的范围。

(10) A. 地理标志权　　B. 物权　　C. 邻接权　　D. 商业秘密权

●某开发人员不顾企业有关保守商业秘密的要求，将其参与该企业开发设计的应用软件的核心程序设计技巧和算法通过论文向社会发表，那么该开发人员的行为__(11)__。

(11) A. 属于开发人员权利，不涉及企业权利　　B. 侵犯了企业商业秘密权

C. 违反了企业的规章制度但不侵权　　D. 未侵犯权利人软件著作权

●W3C 制定了同步多媒体集成语言规范，称为__(12)__规范。

(12) A. XML　　B. SMIL　　C. VRML　　D. SGML

●以下显示器像素点距的规格中，最好的是__(13)__。

(13) A. 0.39　　B. 0.33　　C. 0.31　　D. 0.28

●800×600 的分辨率的图像，若每个像素具有 16 位的颜色深度，则可表示__(14)__种不同的颜色。

(14) A. 1000　　B. 1024　　C. 65 536　　D. 480 000

●CVS 是一种__(15)__工具。

(15) A. 需求分析　　B. 编译　　C. 程序编码　　D. 版本控制

●通常在软件的__(16)__活动中无须用户参与。

(16) A. 需求分析　　B. 维护　　C. 编码　　D. 测试

●软件能力成熟度模型(CMM)是目前国际上最流行、最实用的软件生产过程标准和软件企业成熟度的等级认证标准。该模型将软件能力成熟度自低到高依次划分为初始级、可重复级、已定义级、已管理级、优化级。从__(17)__开始，要求企业建立基本的项目管理过程的政策和管理规程，使项目管理工作有章可循。

(17) A. 初始级　　B. 可重复级　　C. 已定义级　　D. 已管理级

●在软件开发中，__(18)__不能用来描述项目开发的进度安排。在其他三种图中，可用__(19)__动态地反映项目开发进展情况。

(18)A. 甘特图　　B. PERT 图　　C. PERT/CPM 图　　D. 鱼骨图

(19)A. 甘特图　　B. PERT 图　　C. PERT/CPM 图　　D. 鱼骨图

●C 语言是一种__(20)__语言。

(20) A. 编译型　　B. 解释型　　C. 编译、解释混合型　　D. 脚本

●若程序运行时系统报告除数为 0，这属于__(21)__错误。

(21) A. 语法　　B. 语用　　C. 语义　　D. 语境

●集合 $L=\{a^m b^m \mid m \geqslant 0\}$ __(22)__。

(22) A. 可用正规式“a^*b^*”表示

B. 不能用正规式表示，但可用非确定的有限自动机识别

C. 可用正规式“$a^m b^m$”表示

D. 不能用正规式表示，但可用上下文无关文法表示

●设备驱动程序是直接与__(23)__打交道的软件模块。一般而言，设备驱动程序的任务是接收来自与设备__(24)__。

(23) A. 硬件　　B. 办公软件　　C. 编译程序　　D. 连接程序

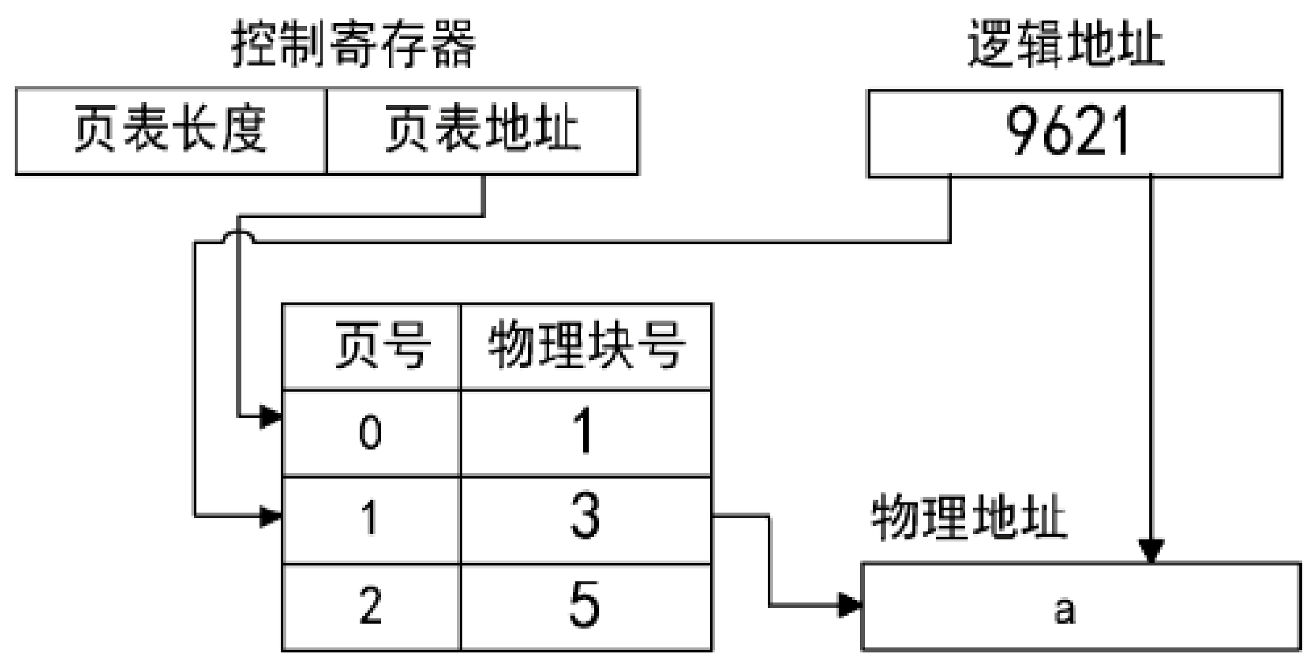
(24) A. 有关的上层软件的抽象请求，进行与设备相关的处理

B. 无关的上层软件的抽象请求，进行与设备相关的处理

C. 有关的上层软件的抽象请求，进行与设备无关的处理

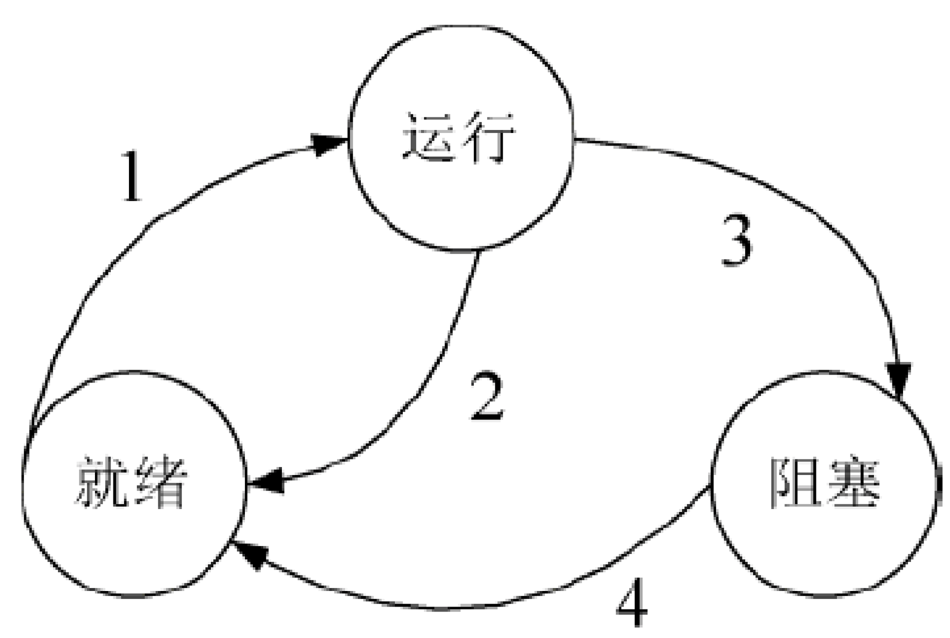
D. 无关的上层软件的抽象请求，进行与设备无关的处理

●页式存储系统的逻辑地址是由页号和页内地址两部分组成，地址变换过程如下图所示。假定页面的大小为 8KB，图中所示的十进制逻辑地址 9621，经过地址变换后，形成的物理地址 a 应为十进制 (25) 。



(25) A. 42 380　　B. 25 996　　C. 9612　　D. 8192

●某系统的进程状态转换如下图所示，图中 1、2、3 和 4 分别表示引起状态转换的不同原因，原因 4 表示 (26) ；一个进程状态转换会引起另一个进程状态转换的是 (27) 。



(26) A. 就绪进程被调度　　B. 运行进程执行了 P 操作

C. 发生了阻塞进程等待的事件　　D. 运行进程的时间片到了

(27) A. 1→2　　B. 2→1　　C. 3→2　　D. 2→4

●在操作系统中，虚拟设备通常采用 (28) 设备来提供虚拟设备。

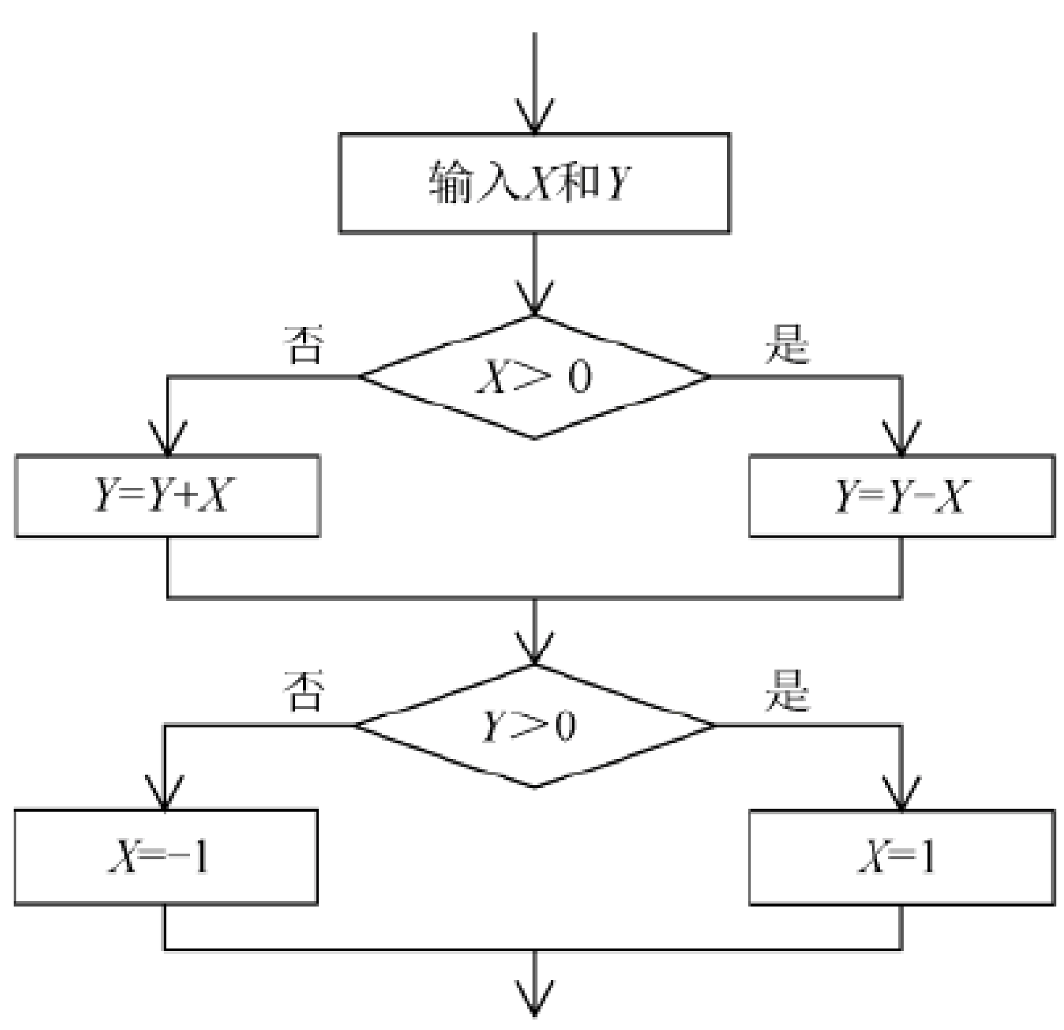
(28) A. Spooling 技术，利用磁带　　B. Spooling 技术，利用磁盘

C. 脱机批处理技术，利用磁盘　　D. 通道技术，利用磁带

●某软件在应用初期运行在 Windows NT 环境中。现因某种原因，该软件需要在 UNIX 环境中运行，而且必须完成相同的功能。为适应这个要求，软件本身需要进行修改，而所需修改的工作量取决于该软件的 (29) 。

(29) A. 可扩充性　　B. 可靠性　　C. 复用性　　D. 可移植性

●对于如下的程序流程，当采用语句覆盖法设计测试案例时，至少需要设计 (30) 个测试案例。



(30) A. 1　　B. 2　　C. 3　　D. 4

●内聚性和耦合性是度量软件模块独立性的重要准则，软件设计时应力求__(31)__。

(31) A. 高内聚，高耦合　　B. 高内聚，低耦合
C. 低内聚，高耦合　　D. 低内聚，低耦合

●统一过程(UP)是一种用例驱动的迭代式增量开发过程，每次迭代过程中主要的工作流包括捕获需求、分析、设计、实现和测试等。这种软件过程的用例图(Use Case Diagram)是通过__(32)__得到的。

(32) A. 捕获需求　　B. 分析　　C. 设计　　D. 实现

●在某大学学生学籍管理信息系统中，假设学生年龄的输入范围为 16～40，则根据黑盒测试中的等价类划分技术，下面划分正确的是__(33)__。

(33) A. 可划分为 2 个有效等价类，2 个无效等价类
B. 可划分为 1 个有效等价类，2 个无效等价类
C. 可划分为 2 个有效等价类，1 个无效等价类
D. 可划分为 1 个有效等价类，1 个无效等价类

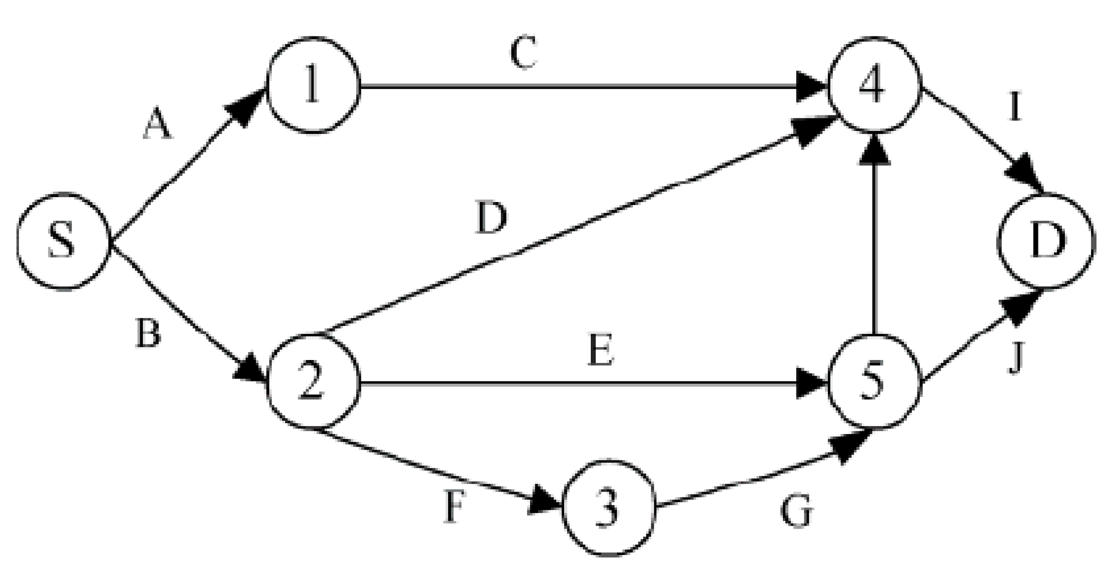
●软件__(34)__的提高，有利于软件可靠性的提高。

(34) A. 存储效率　　B. 执行效率　　C. 容错性　　D. 可移植性

●正式的技术评审(Formal Technical Review，FTR)是软件工程师组织的软件质量保证活动，下面关于 FTR 指导原则中不正确的是__(35)__。

(35) A. 评审产品，而不是评审生产者的能力
B. 要有严格的评审计划，并遵守日程安排
C. 对评审中出现的问题要充分讨论，以求彻底解决
D. 限制参与者人数，并要求评审会之前做好准备

●某工程计划如下图所示，各个作业所需的天数如下表所示，设该工程从第 0 天开工，则该工程的最短工期是__(36)__天。



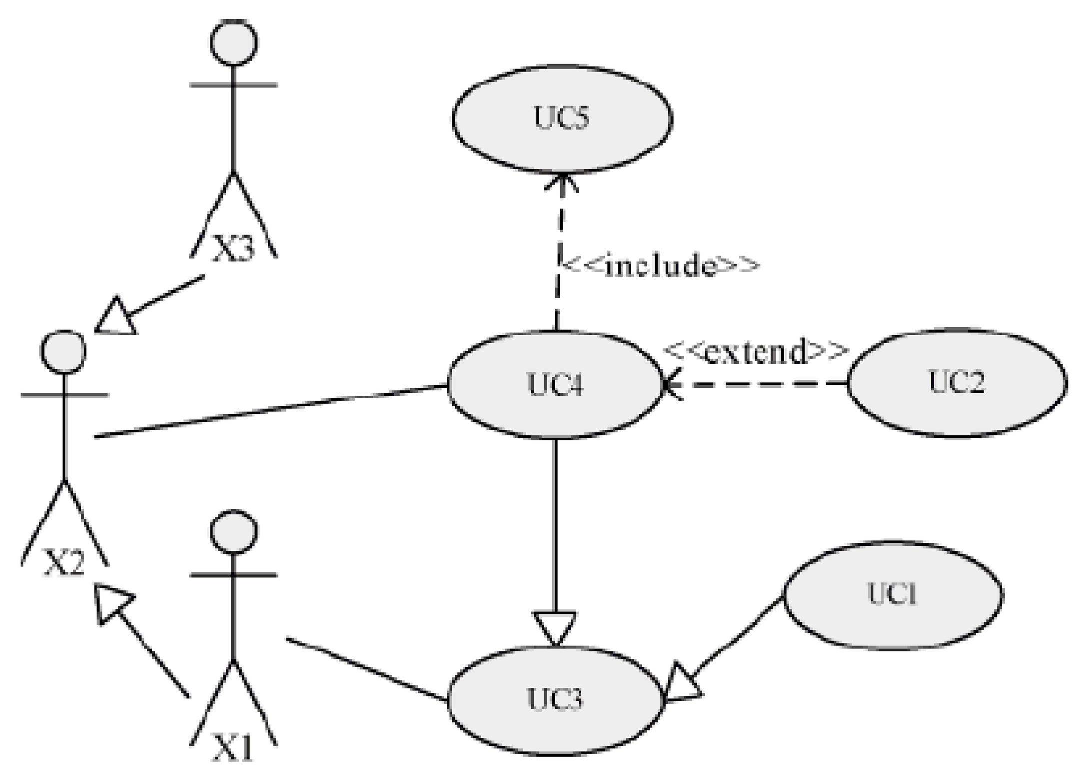
作业	A	B	C	D	E	F	G	H	I	J
所需天数	7	6	8	10	7	3	2	4	3	7

(36) A. 17　　B. 18　　C. 19　　D. 20

●__(37)__表示了系统与参与者之间的接口。在每一个用例中，该对象从参与者处收集信息，并将之转换为一种被实体对象和控制对象使用的形式。

(37) A. 边界对象　　B. 可视化　　C. 抽象对象　　D. 实体对象

●在下面的用例图中，X1、X2 和 X3 表示__(38)__，已知 UC3 是抽象用例，那么 X1 可通过__(39)__用例与系统进行交互。并且，用例__(40)__是 UC4 的可选部分，用例__(41)__是 UC4 的必需部分。



(38) A. 人　　B. 系统　　C. 参与者　　D. 外部软件

(39) A. UC4、UC1　　B. UC5、UC1　　C. UC5、UC2　　D. UC1、UC2

(40)A. UC1　　B. UC2　　C. UC3　　D. UC5

(41)A. UC1　　B. UC2　　C. UC3　　D. UC5

●在统一建模语言(UML)中，(42) 用于描述系统与外部系统及用户之间的交互。

(42)A. 类图　　B. 用例图　　C. 对象图　　D. 协作图

●面向对象分析与设计是面向对象软件开发过程中的两个重要阶段，下列活动中，(43) 不属于面向对象分析阶段。

(43)A. 构建分析模型　　B. 识别分析类　　C. 确定接口规格　　D. 评估分析模型

●在“模型－视图－控制器”(MVC)模式中，(44) 主要表现用户界面，(45) 用来描述核心业务逻辑。

(44)A. 视图　　B. 模型　　C. 控制器　　D. 视图和控制器

(45)A. 视图　　B. 模型　　C. 控制器　　D. 视图和控制器

●当采用标准 UML 构建系统类模型(Class Model)时，若类 B 除具有类 A 的全部特性外，类 B 还可定义新的特性以及置换类 A 的部分特性，那么类 B 与类 A 具有 (46) 关系；若类 A 的对象维持类 B 对象的引用或指针，并可与类 C 的对象共享相同的类 B 的对象，那么类 A 与类 B 具有 (47) 关系。

(46)A. 聚合　　B. 泛化　　C. 传递　　D. 迭代

(47)A. 聚合　　B. 泛化　　C. 传递　　D. 迭代

●序言性注释是指在每个程序或模块开头的一段说明，起辅助理解程序的作用，一般包括：程序的表示、名称和版本号，程序功能描述，接口与界面描述，输入/输出数据说明，开发历史，与运行环境有关的信息等。下列叙述中不属于序言性注释的是 (48) 。

(48) A. 程序对硬件、软件资源的要求
B. 重要变量和参数说明
C. 嵌入在程序中的 SQL 语句
D. 程序开发的原作者、审查者、修改者、编程日期等

●程序设计语言中，(49) 。

(49) A. while 循环语句的执行效率比 do-while 循环语句的执行效率高
B. while 循环语句的循环体执行次数比循环条件的判断次数多 1，而 do-while 语句的循环体执行次数比循环条件的判断次数少 1
C. while 语句的循环体执行次数比循环条件的判断次数少 1，而 do-while 语句的循环体执行次数比循环条件的判断次数多 1
D. while 语句的循环体执行次数比循环条件的判断次数少 1，而 do-while 语句的循环体执行次数等

于循环条件的判断次数

●正则表达式　$1^*(0|01)^*$ 表示的集合元素的特点是 (50) 。

(50) A. 长度为奇数的 0、1 串　　B. 开始和结尾字符必须为 1 的 0、1 串

C. 串的长度为偶数的 0、1 串　　D. 不包含子串 011 的 0、1 串

●在数据库管理系统中， (51) 不属于安全性控制机制。

(51) A. 完整性约束　　B. 视图　　C. 密码验证　　D. 用户授权

●设关系模式 R(A, B, C)，传递依赖指的是 (52) ；下列结论错误的是 (53) 。

(52) A. 若 A→B，B→C，则 A→C　　B. 若 A→B，A→C，则 A→BC

C. 若 A→C，则 AB→C　　D. 若 A→BC，则 A→B，A→C

(53) A. 若 A→BC，则 A→B，A→C　　B. 若 A→B，A→C，则 A→BC

C. 若 A→C，则 AB→C　　D. 若 AB→C，则 A→C，B→C

●建立一个供应商、零件数据库。其中“供应商”表 S(Sno, Sname, Zip, City)分别表示供应商代码、供应商名、供应商邮编、供应商所在城市，其函数依赖为：Sno→(Sname, Zip, City)，Zip→City。“供应商”表 S 属于 (54) 。

(54) A. 1NF　　B. 2NF　　C. 3NF　　D. BCNF

●关系 R、S 如下图所示，R ⋈ S 可由 (55) 基本的关系运算组成，R ⋈ S = (56) 。

关系 R

A	B	C
a	b	c
b	a	d
c	d	e
d	f	g

关系 S

A	C	D
a	c	d
d	f	g
b	d	g

(55) A. π、σ和×　　B. −、σ和×

C. ∩、σ和×　　D. π、σ和∩

(56) A.

A	B	C
a	b	c
b	a	d
c	d	e

B.

A	B	C	D
a	b	c	d
b	a	d	g
d	f	g	g

C.

A	B	C
a	b	c
b	a	d

D.

A	B	C	D
a	b	c	d
b	a	d	g

●设栈 S 和队列 Q 的初始状态为空，元素按照 a、b、c、d、e 的次序进入栈 S，当一个元素从栈中出来后立即进入队列 Q。若队列的输出元素序列是 c、d、b、a、e，则元素的出栈顺序是 (57) ，栈 S 的容量至少为 (58) 。

(57) A. a、b、c、d、e　　B. e、d、c、b、a

C. c、d、b、a、e　　D. e、a、b、d、c

(58) A. 2　　B. 3　　C. 4　　D. 5

●输入受限的双端队列是指元素只能从队列的一端输入，但可以从队列的两端输出，如下图所示。若有 8、1、4、2 依次进入输入受限的双端队列，则得不到输出序列 (59) 。



(59)A. 2、8、1、4　　B. 1、4、8、2　　C. 4、2、1、8　　D. 2、1、4、8

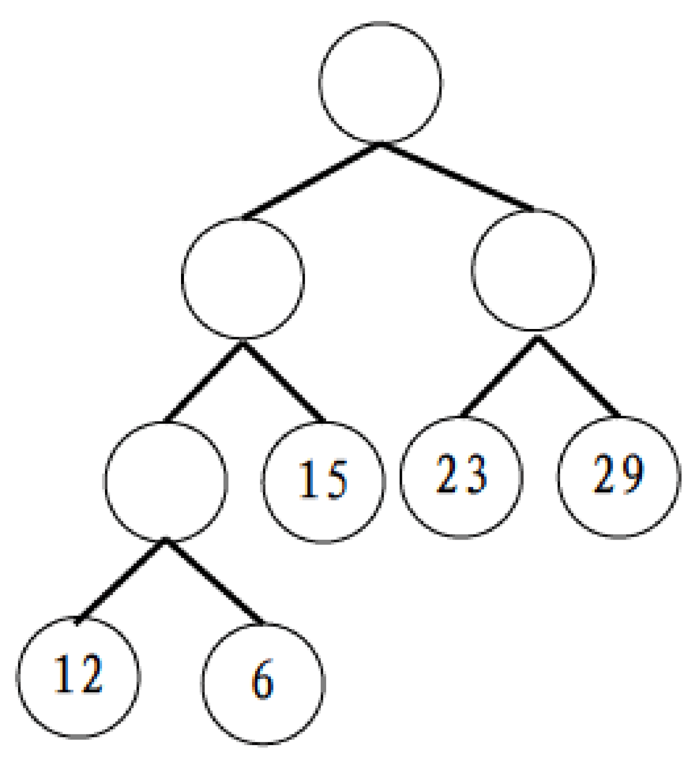
●对于 $n(n\geqslant0)$个元素构成的线性序列 L，在 (60) 时适合采用链式存储结构。

(60) A. 需要频繁修改 L 中元素的值　　B. 需要频繁地对 L 进行随机查找

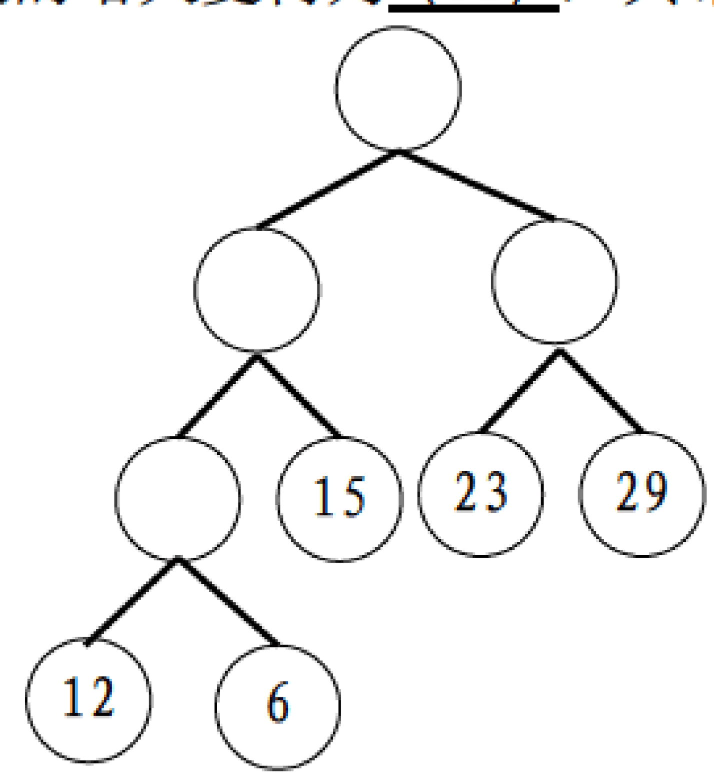
C. 需要频繁地对 L 进行删除和插入操作　　D. 要求 L 存储密度高

●由权值为 29、12、15、6、23 的五个叶子结点构造的哈夫曼树为 (61) ，其带权路径长度为 (62) 。

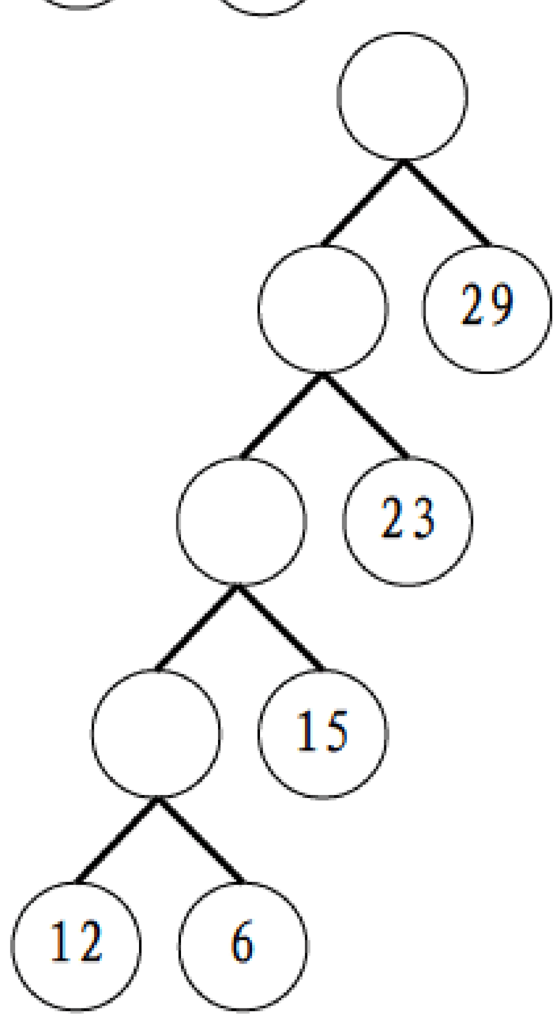
(61)A.



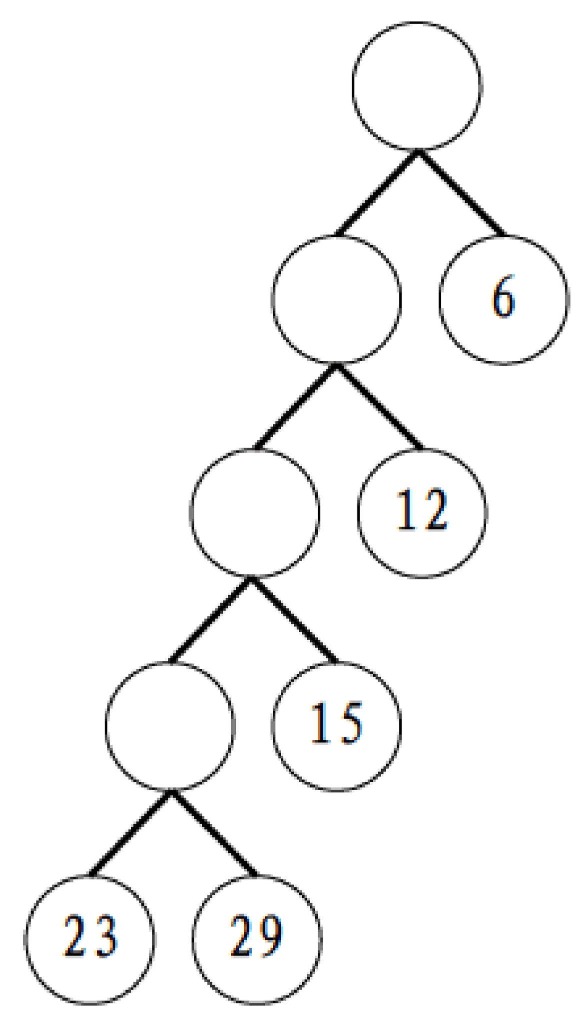
B.



C.



D.



(62) A. 85　　B. 188　　C. 192　　D. 222

●已知某二叉树的中序列为 CBDAEFI、先序列为 ABCDEFI，则该二叉树的高度为 (63) 。

(63) A. 2　　B. 3　　C. 4　　D. 5

●若一个问题既可以用迭代方式也可以用递归方式求解，则 (64) 方法具有更高的时空效率。

(64) A. 迭代　　B. 递归　　C. 先递归后迭代　　D. 先迭代后递归

●设商店有 10 元、5 元、2 元和 1 元的零币，每种零币数量充足。售货员给顾客找零钱时，零币的数量越少越好。例如给顾客找零 29 元：先选 2 张 10 元币，然后选择 1 张 5 元币，再选择两张 2 元币。以上的找零钱方法采用了 (65) 策略。

(65)A. 分治　　B. 贪心　　C. 动态规划　　D. 回溯

●分配给某校园网的地址块是 202.105.192.0/18，该校园网包含 (66) 个 C 类网络。

(66) A. 6　　B. 14　　C. 30　　D. 62

●以下关于 DHCP 服务的说法中正确的是 (67) 。

(67) A. 在一个子网内只能设置一台 DHCP 服务器，以防止冲突

B. 在默认情况下，客户机采用最先到达的 DHCP 服务器分配的 IP 地址

C. 使用 DHCP 服务，无法保证某台计算机使用固定 IP 地址

D. 客户端在配置时必须指明 DHCP 服务器的 IP 地址，才能获得 DHCP 服务

●电子邮件应用程序利用 POP3 (68) 。

(68) A. 创建邮件　　B. 加密邮件　　C. 发送邮件　　D. 接收邮件

●“<title style="italic">science</title>”是一个 XML 元素的定义，其中元素标记的属性值是 (69) 。

(69) A. title　　B. style　　C. italic　　D. science

●与多模光纤相比较，单模光纤具有 (70) 等特点。

(70) A. 较高的传输率、较长的传输距离、较高的成本

B. 较低的传输率、较短的传输距离、较高的成本

C. 较高的传输率、较短的传输距离、较低的成本

D. 较低的传输率、较长的传输距离、较低的成本

●The Rational Unified Process (RUP) is a software engineering process, which captures many of best practices in modern software development. The notions of (71) and scenarios have been proven to be an excellent way to capture function requirements. RUP can be described in two dimensions - time and content. In the time dimension, the software lifecycle is broken into cycles. Each cycle is divided into four consecutive (72) which is concluded with a well-defined (73) and can be further broken down into (74) - a complete development loop resulting in a release of an executable product, a subset of the final product under development, which grows incrementally to become the final system. The content structure refers to the disciplines, which group (75) logically by nature.

(71) A. artifacts　　B. use-cases　　C. actors　　D. workers

(72) A. orientations　　B. views　　C. aspects　　D. phases

(73) A. milestone　　B. end-mark　　C. measure　　D. criteria

(74) A. rounds　　B. loops　　C. iterations　　D. circularities

(75) A. functions　　B. workflows　　C. actions　　D. activities

下午科目

说明：试题五、试题六选做一题。

试题一

阅读以下说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某房屋租赁公司欲建立一个房屋租赁服务系统，统一管理房主和租赁者的信息，从而快速地提供租赁服务。该系统具有以下功能。

(1) 登记房主信息。对于每名房主，系统需登记其姓名、住址和联系电话，并将这些信息写入房主信息文件。

(2) 登记房屋信息。所有在系统中登记的房屋都有一个唯一的识别号(对于新增加的房屋，系统会自动为其分配一个识别号)。除此之外，还需登记该房屋的地址、房型(如平房、带阳台的楼房、独立式住宅等)、

最多能够容纳的房客数、租金及房屋状况(待租赁、已出租)。这些信息都保存在房屋信息文件中。一名房主可以在系统中登记多个待租赁的房屋。

(3) 登记租赁者信息。所有想通过该系统租赁房屋的租赁者，必须首先在系统中登记个人信息，包括姓名、住址、电话号码、出生年月和性别。这些信息都保存在租赁者信息文件中。

(4) 租赁房屋。已经登记在系统中的租赁者，可以得到一份系统提供的待租赁房屋列表。一旦租赁者从中找到合适的房屋，就可以提出看房请求。系统会安排租赁者与房主见面。对于每次看房，系统会生成一条看房记录并将其写入看房记录文件中。

(5) 收取手续费。房主登记完房屋后，系统会生成一份费用单，房主根据费用单缴纳相应的费用。

(6) 变更房屋状态。当租赁者与房主达成租房或退房协议后，房主向系统提交变更房屋状态的请求。系统将根据房主的请求，修改房屋信息文件。

图 1 和图 2 分别给出了该系统的顶层数据流图和 0 层数据流图。

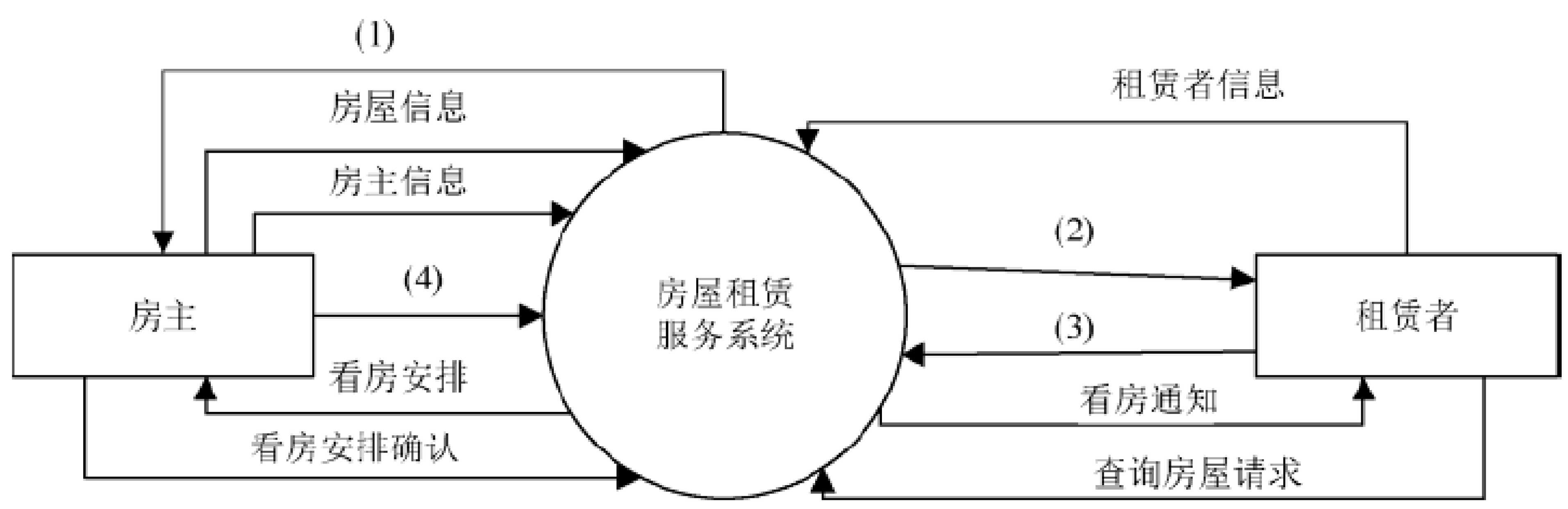


图 1 顶层数据流图

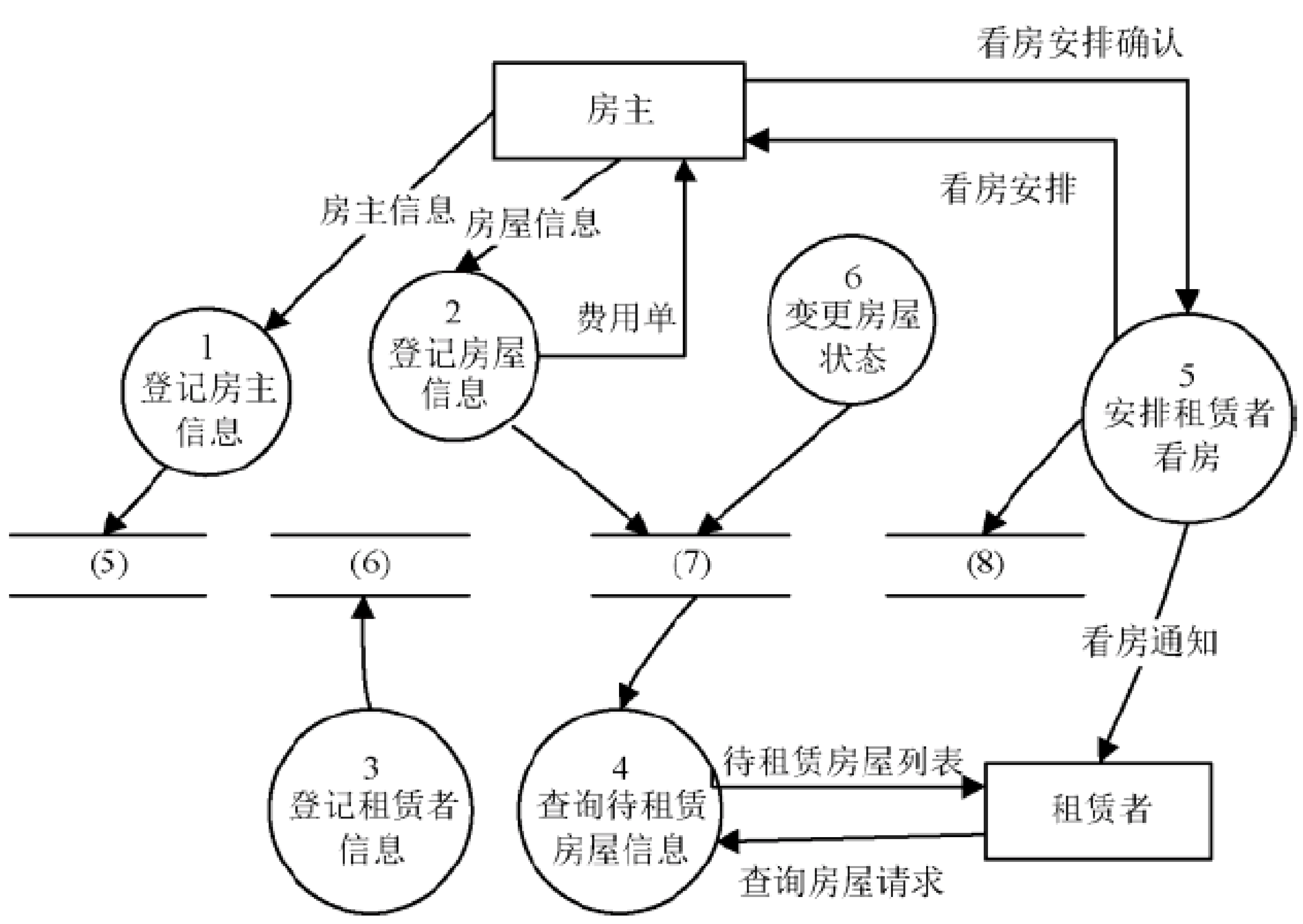


图 2 0 层数据流图

【问题 1】

使用说明中给出的词汇，将图 1 中(1)～(4)处的数据流补充完整。

【问题 2】

使用说明中给出的词汇，将图 2 中的(5)～(8)补充完整。

【问题 3】

图 2 中缺失了三条数据流，指出这三条数据流的起点、终点和数据流名称。

试题二

阅读下列说明，回答问题 1 至问题 4，将解答填入答题纸的对应栏内。

【说明】

某汽车维修站拟开发一套小型汽车维修管理系统，对车辆的维修情况进行管理。

(1) 对于新客户及车辆，汽车维修管理系统首先登记客户信息，包括客户编号、客户名称、客户性质(个人、单位)、折扣率、联系人、联系电话等；还要记录客户的车辆信息，包括车牌号、车型、颜色、车辆类别等。一个客户至少有一台车。客户及车辆信息如表 1 所示。

表 1　客户及车辆信息

客户编号	GS0051	客户名称	××公司	客户性质	单位
折扣率	95%	联系人	杨浩东	联系电话	82638779
车牌号		颜色	车型	车辆类别	
**0765		白色	帕萨特	微型车	

(2) 记录维修车辆的故障信息，包括维修类型(普通、加急)、作业分类(大、中、小修)、结算方式(自付、三包、索赔)等。维修厂的员工分为维修员和业务员。车辆维修首先委托给业务员。业务员对车辆进行检查和故障分析后，与客户磋商，确定故障现象，生成维修委托书，如表 2 所示。

表 2　维修委托书

No.20110702003　　　　登记日期：2011-07-02

车牌号	**0765	客户编号	GS0051	维修类型	普通
作业分类	中修	结算方式	自付	进厂时间	20110702 11:09
业务员	张小江	业务员编号	012	预计完工时间	
故障描述					
车头损坏，水箱漏水					

(3) 维修车间根据维修委托书和车辆的故障现象，在已有的维修项目中选择并确定一个或多个具体维修项目，安排相关的维修工及工时，生成维修派工单。维修派工单如表 3 所示。

表 3　维修派工单

No.20110702003

维修项目编号	维修项目	工时	维修员编号	维修员工种
012	维修车头	5.00	012	机修
012	维修车头	2.00	023	漆工
015	水箱焊接补漏	1.00	006	焊工
017	更换车灯	1.00	012	机修

(4) 客户车辆在车间修理完毕后，根据维修项目单价和维修派工单中的工时计算车辆此次维修的总费用，记录在委托书中。

根据需求阶段收集的信息，设计的实体联系图(见图 1)和关系模式(不完整)如下所示。图 1 中业务员和维修工是员工的子实体。

【概念结构设计】

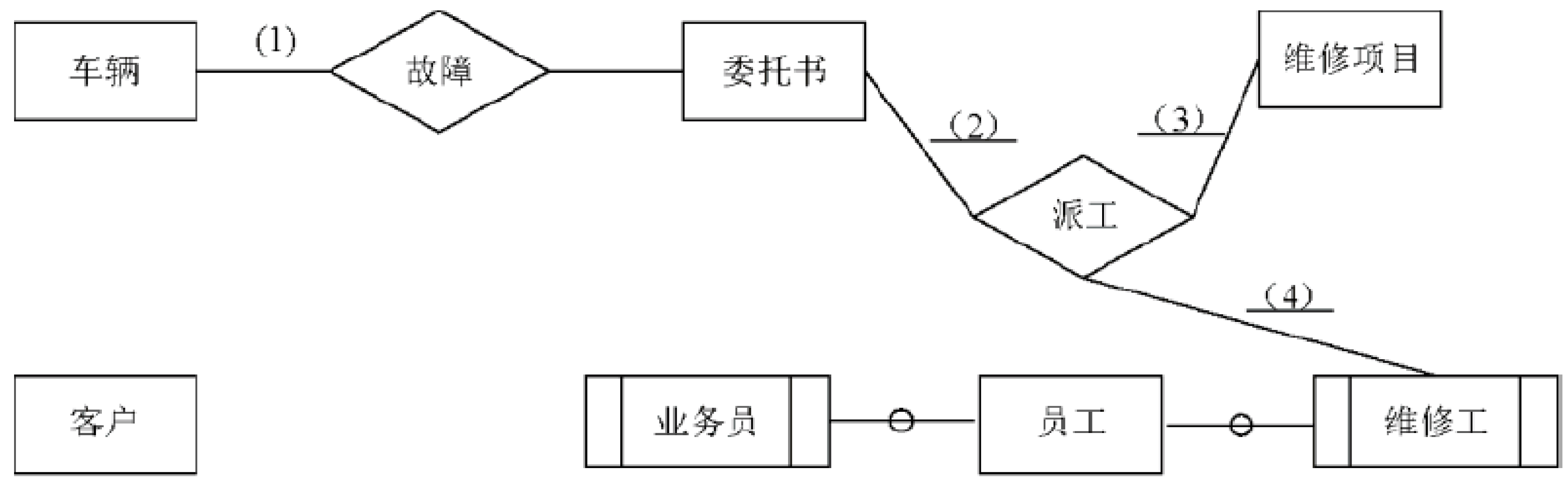


图 1　实体联系图

【逻辑结构设计】

客户(______(5)______，折扣率，联系人，联系电话)

车辆(车牌号，客户编号，车型，颜色，车辆类别)

委托书(______(6)______，维修类型，作业分类，结算方式，进厂时间，预计完工时间，登记日期，故障描述，总费用)

维修项目(维修项目编号，维修项目，单价)

派工单(　　(7)　　，工时)

员工(______(8)______，工种，员工类型，级别)

【问题 1】

根据问题描述，填写图 1 中(1)～(4)处联系的类型。联系类型分为一对一、一对多和多对多三种，分别使用 1∶1、1∶n 或 1∶*、m∶n 或*∶*表示。

【问题 2】

补充图 1 中的联系并指明其联系类型。联系名可为联系 1，联系 2，…

【问题 3】

根据图 1 和说明，将逻辑结构设计阶段生成的关系模式中的空(5)～(8)补充完整。

【问题 4】

根据问题描述，写出客户、委托书和派工单这三个关系的主键。

试题三

阅读下列说明和图，回答问题 1 至问题 3，将解答填入答题纸的对应栏内。

【说明】

某图书管理系统的主要功能如下。

(1)　图书管理系统的资源目录中记录着所有可供读者借阅的资源，每项资源都有一个唯一的索引号。系统需登记每项资源的名称、出版时间和资源状态(可借阅或已借出)。

(2)　资源可以分为两类：图书和唱片。对于图书，系统还需登记作者和页数；对于唱片，还需登记演唱者和介质类型(CD 或者磁带)。

(3)　读者信息保存在图书管理系统的读者信息数据库中，记录的信息包括读者的识别码和读者姓名。系统为每个读者创建了一个借书记录文件，用来保存读者所借资源的相关信息。

现采用面向对象方法开发该图书管理系统。识别类是面向对象分析的第一步。比较常用的识别类的方法是寻找问题描述中的名词，再根据相关规则从这些名词中删除不可能成为类的名词，最终得到构成该系

统的类。表 1 给出了说明中出现的所有名词。

表 1　图书管理系统中的名词

图书管理系统	资源目录	读　者	资　源
索引号	系统	名称	出版时间
资源状态	图书	唱片	作者
页数	演唱者	介质类型	CD
磁带	读者信息	读者信息数据库	识别码
姓名	借书记录文件	信息	

通过对表 1 中的名词进行分析，最终得到了图 1 所示的 UML 类图(类的说明见表 2)。

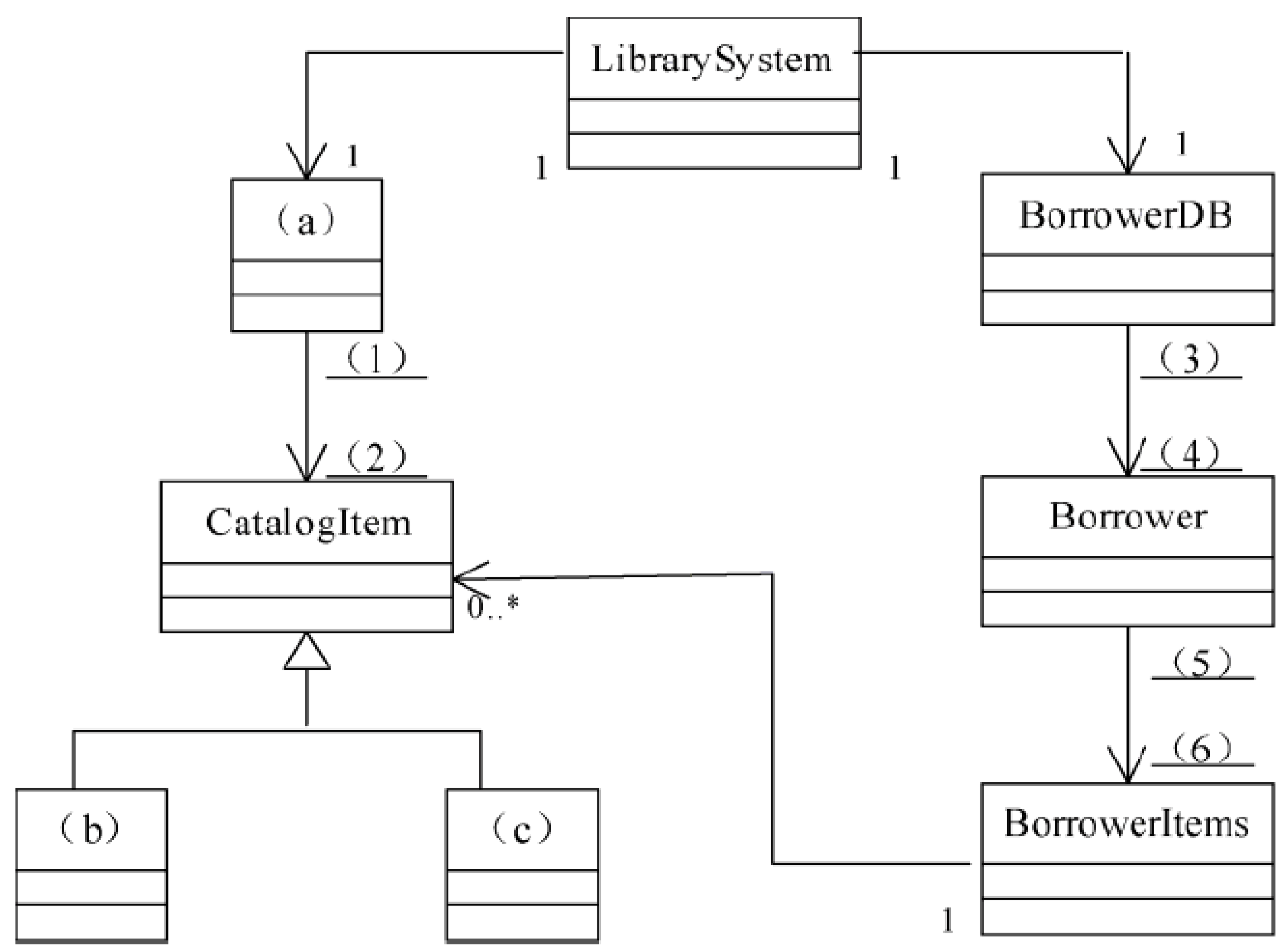


图 1　UML 类图

表 2　类说明

类　名	说　明
LibrarySystem	图书管理系统
BorrowerDB	保存读者信息的数据库
CatalogItem	资源目录中保存的每项资源
Borrower	读者
BorrowerItems	为每个读者创建的借书记录文件

【问题 1】

表 2 所给出的类并不完整，根据说明和表 2，将图 1 中的(a)～(c)处补充完整。

【问题 2】

根据说明中的描述，给出图 1 中的类 CatalogItem 以及(b)、(c)处所对应的类的关键属性(使用表 1 中给出的词汇)，其中，CatalogItem 有 4 个关键属性；(b)、(c)处对应的类各有 2 个关键属性。

【问题 3】

识别关联的多重度是面向对象建模过程中的一个重要步骤。根据说明中给出的描述，完成图 1 中的(1)～(6)。

试题四

阅读以下说明、图和 C 代码，将应填入 *(n)* 处的字句写在答题纸的对应栏内。

【说明】

一般的树结构常采用孩子-兄弟表示法表示，即用二叉链表作树的存储结构，链表中结点的两个链域分别指向该结点的第一个孩子结点和下一个兄弟结点。例如，图 1(a)所示的树的孩子-兄弟表示如图 1(b)所示。

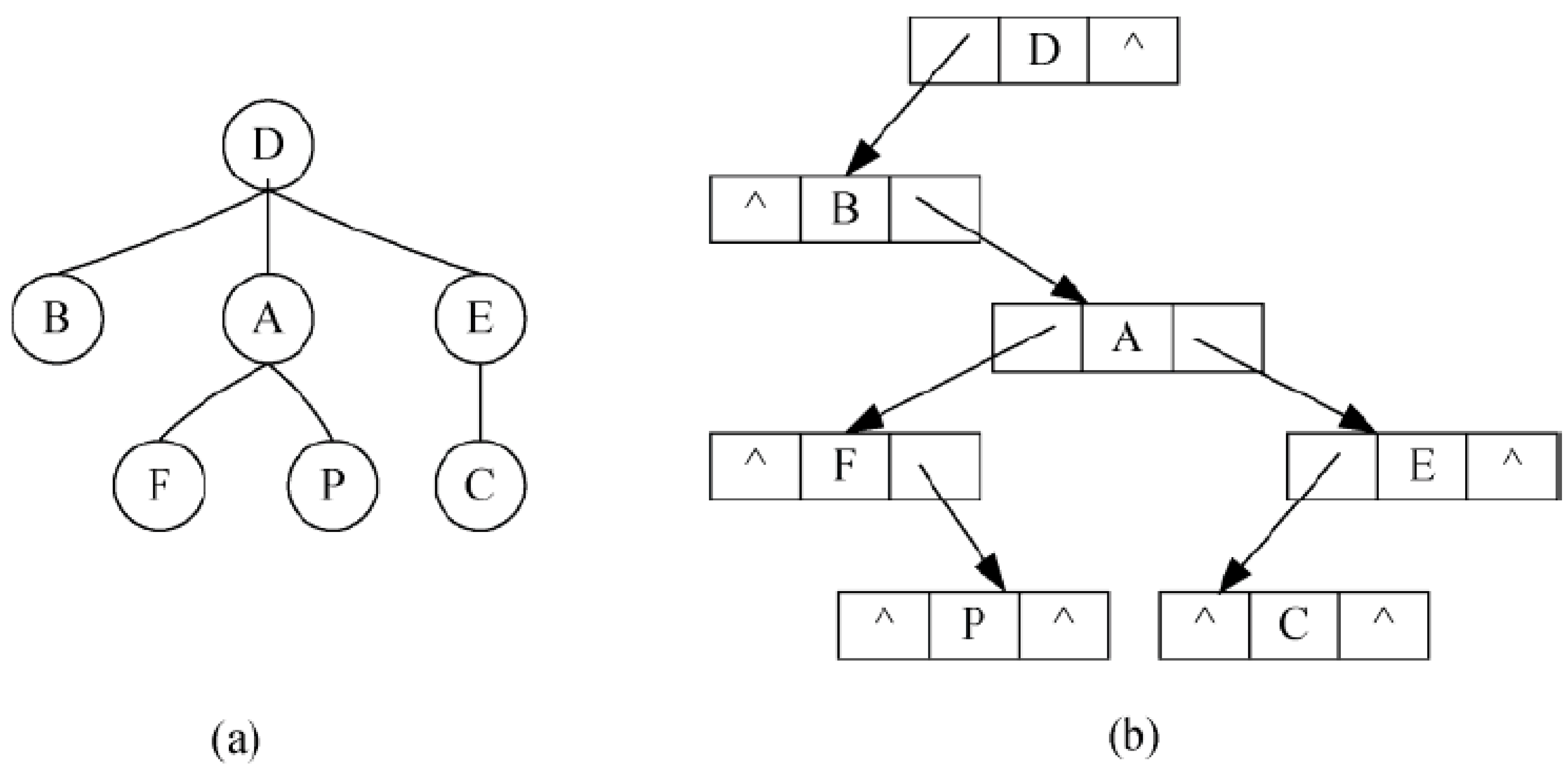


图 1 树及其孩子-兄弟表示示意图

函数 LevelTraverse()的功能是对给定树进行层序遍历。例如，对图 1 所示的树进行层序遍历时，结点的访问次序为 D B A E F P C 。

对树进行层序遍历时使用了队列结构，实现队列基本操作的函数原型如表 1 所示。

表 1 实现队列基本操作的函数原型

函数原型	说　明
void InitQueue(Queue *Q)	初始化队列
Bool IsEmpty(Queue Q)	判断队列是否为空，若是则返回 TRUE，否则返回 FALSE
void EnQueue(Queue *Q,TreeNode p)	元素入队列
void DeQueue(Queue *Q,TreeNode *p)	元素出队列

Bool、Status 类型定义如下：

```
typedef enum {FALSE = 0,TRUE = 1} Bool;
typedef enum {OVERFLOW = -2,UNDERFLOW = -1,ERROR = 0,OK = 1} Status;
```

树的二叉链表结点定义如下：

```
typedef struct Node {
  char data;
  struct Node *firstchild,*nextbrother;
}Node,*TreeNode;
```

【C 代码】

```
Status LevelTraverse(TreeNode root)
{  /*层序遍历树，树采用孩子-兄弟表示法，root 是树根结点的指针*/
  Queue tempQ;
  TreeNode ptr,brotherptr;
  if (!root)
    return ERROR;
  InitQueue(&tempQ);
  (1) ;
  brotherptr = root -> nextbrother;
  while (brotherptr){ EnQueue(&tempQ,brotherptr);
    (2) ;
  }  /*end-while*/
  while ( (3) ) {
    (4) ;
    printf("%c\t",ptr->data);
    if ( (5) ) continue;
    (6) ;
    brotherptr = ptr->firstchild->nextbrother;
    while (brotherptr){ EnQueue(&tempQ,brotherptr);
      (7) ;
    }  /*end-while*/
  }  /*end-while*/
  return OK;
}/*LevelTraverse*/
```

试题五

阅读下列说明和 C++代码，将应填入 (n) 处的字句写在答题纸的对应栏内。

【说明】

某游戏公司现欲开发一款面向儿童的模拟游戏，该游戏主要模拟现实世界中各种鸭子的发声特征、飞行特征和外观特征。游戏需要模拟的鸭子种类及其特征如表 1 所示。

表 1　游戏需要模拟的鸭子种类及其特征

鸭子种类	发声特征	飞行特征	外观特征
灰鸭(MallardDuck)	发出“嘎嘎”声(Quack)	用翅膀飞行(FlyWithWings)	灰色羽毛
红头鸭(RedHeadDuck)	发出“嘎嘎”声(Quack)	用翅膀飞行(FlyWithWings)	灰色羽毛、头部红色
棉花鸭(CottonDuck)	不发声(QuackNoWay)	不能飞行(FlyNoWay)	白色
橡皮鸭(RubberDuck)	发出橡皮与空气摩擦的声音(Squeak)	不能飞行(FlyNoWay)	黑白橡皮色

为支持将来能够模拟更多种类鸭子的特征，采用策略(Strategy) 模式设计的类图如图 1 所示。

其中，Duck 为抽象类，描述了抽象的鸭子，而类 RubberDuck、MallardDuck、CottonDuck 和 RedHeadDuck 分别描述具体的鸭子种类，方法 fly()、quack()和 display()分别表示不同种类的鸭子都具有飞行特征、发声

特征和外观特征；类 FlyBehavior 与 QuackBehavior 为抽象类，分别用于表示抽象的飞行行为与发声行为；类 FlyNoWay 与 FlyWithWings 分别描述不能飞行的行为和用翅膀飞行的行为；类 Quack、Squeak 与 QuackNoWay 分别描述发出“嘎嘎”声的行为、发出橡皮与空气摩擦声的行为与不发声的行为。请填补以下代码中的空缺。

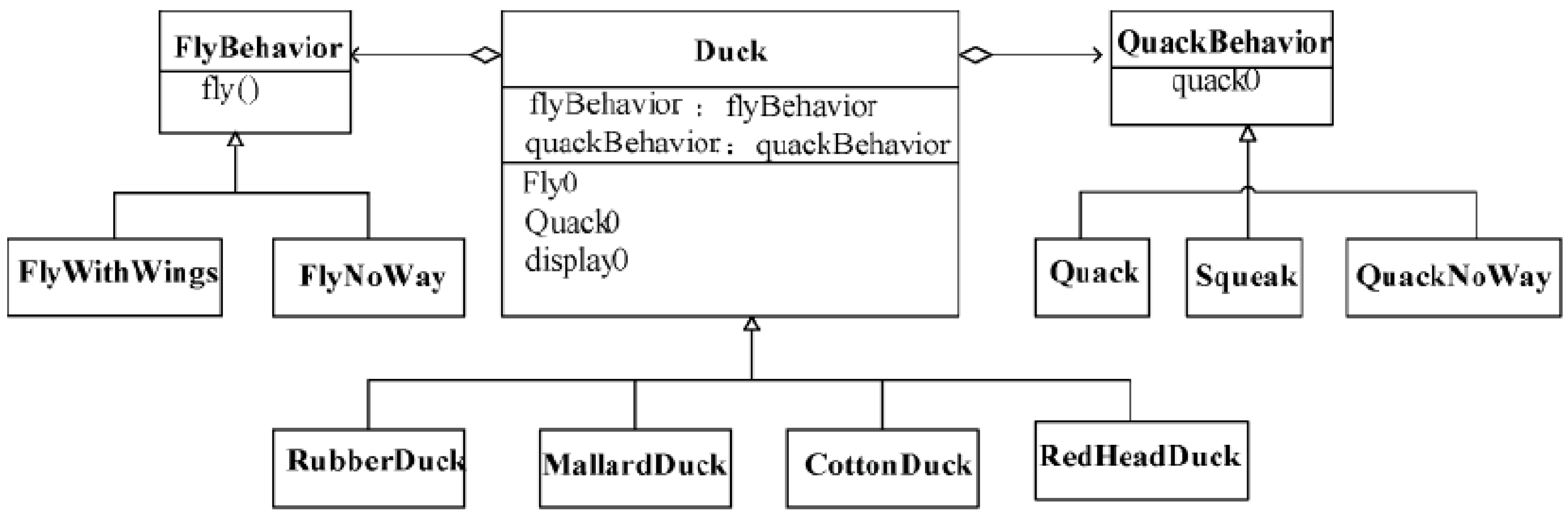


图 1　设计类图

【C++代码】

```
#include<iostream>
    using namespace__(1)__;
    class FlyBehavior{
    public:__(2)__ fly( )=0;
};
class QuackBehavior{
    public:__(3)__quack( )=0;
}

class FlyWithWings :public FlyBehavior{
    public: void fly( ){cout<<"使用翅膀飞行！"<<endl;}
};
class FlyNoWay :public FlyBehavior{
    public: void fly( ){cout<<"不能飞行！"<<endl;}
};
class Quack :public QuackBehavior{
    public: void quack( ){cout<<"发出\'嘎嘎'\声！"<<endl;}
};
class Squeak :public QuackBehavior{
    public: void quack( ){cout<<"发出空气与橡皮摩擦声！"<<endl;}
};
class QuackNoWay :public QuackBehavior{
    public: void quack( ){cout<<"不能发声！"<<endl;}
};
Class Duck{
protected:
    FlyBehavior *__(4)__;
QuackBehavior * __(5)__  ;

public:
void fly(){  __(6)__;  }
void quack() {  __(7)__; };
virtual void display()=0;
};
```

```
class RubberDuck: public Duck {
public:
RubberDuck( ){
flyBehavior=new  (8) ;
    quackBehavior=new  (9) ;
    }
    ~RubberDuck( ){
        if (!flyBehavior) delete flyBehavior;
        if (!quackBehavior) delete quackBehavior;
    }
    void display( ){/*此处省略显示橡皮鸭的代码*/}
//其他代码省略
```

试题六

阅读下列说明和 Java 代码，将应填入 *(n)* 处的字句写在答题纸的对应栏内。

【说明】

某游戏公司现欲开发一款面向儿童的模拟游戏，该游戏主要模拟现实世界中各种鸭子的发声特征、飞行特征和外观特征。游戏需要模拟的鸭子种类及其特征如表 1 所示。

表 1 游戏需要模拟的鸭子种类及其特征

鸭子种类	发声特征	飞行特征	外观特征
灰鸭(MallardDuck)	发出“嘎嘎”声(Quack)	用翅膀飞行(FlyWithWings)	灰色羽毛
红头鸭(RedHeadDuck)	发出“嘎嘎”声(Quack)	用翅膀飞行(FlyWithWings)	灰色羽毛、头部红色
棉花鸭(CottonDuck)	不发声(QuackNoWay)	不能飞行(FlyNoWay)	白色
橡皮鸭(RubberDuck)	发出橡皮与空气摩擦的声音(Squeak)	不能飞行(FlyNoWay)	黑白橡皮色

为支持将来能够模拟更多种类鸭子的特征，采用策略设计(Strategy)模式设计的类图如图 1 所示。

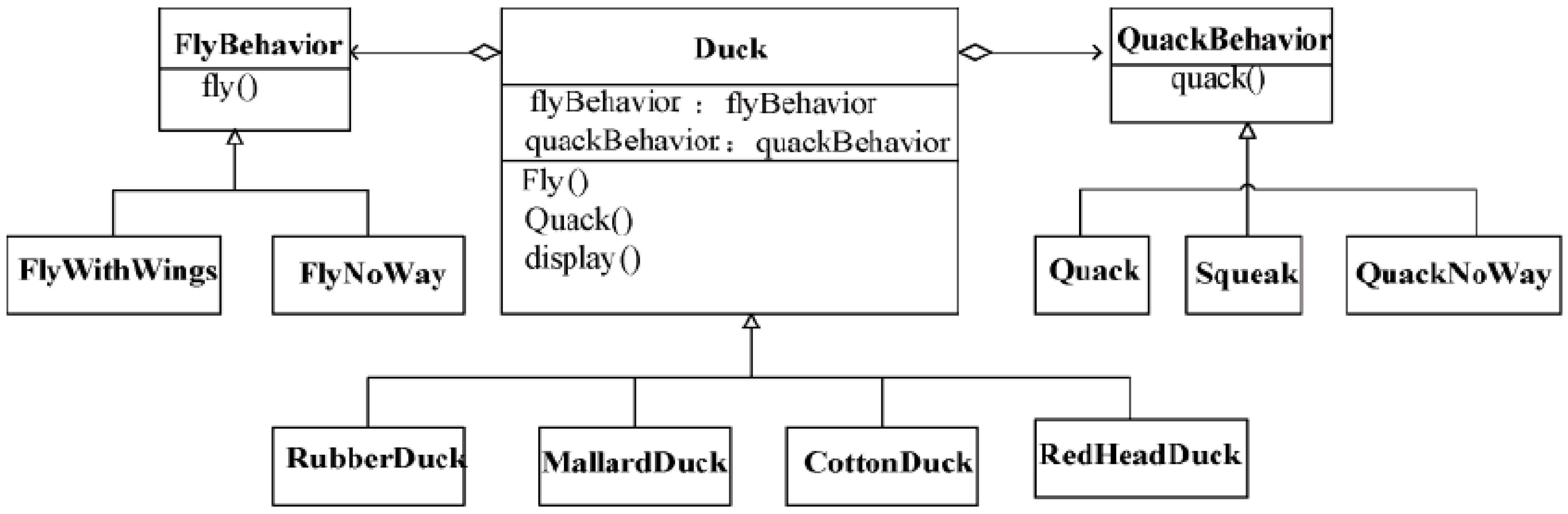


图 1 设计类图

其中，Duck 为抽象类，描述了抽象的鸭子，而类 RubberDuck、MallardDuck、CottonDuck 和 RedHeadDuck 分别描述具体的鸭子种类，方法 fly()、quack()和 display()分别表示不同种类的鸭子都具有飞行特征、发声特征和外观特征；接口 FlyBehavior 与 QuackBehavior 分别用于表示抽象的飞行行为与发声行为；类 FlyNoWay 与 FlyWithWings 分别描述不能飞行的行为和用翅膀飞行的行为；类 Quack、Squeak 与

QuackNoWay 分别描述发出“嘎嘎”声的行为、发出橡皮与空气摩擦声的行为与不发声的行为。请填补以下代码中的空缺。

【Java 代码】

```
 (1) FlyBehavior{
        public void fly( );
};
 (2) QuackBehavior{
        public void quack( );
};
class FlyWithWings implements FlyBehavior{
    public void fly( ){System.out.println("使用翅膀飞行！");}
};
class FlyNoWay implements FlyBehavior{
    public void fly( ){System.out.println("不能飞行！");}
};
class Quack implements QuackBehavior {
    public void quack( ){System.out.println(" 发出\'嘎嘎'\声！");}
};
class Squeak implements QuackBehavior {
    public void quack( ){System.out.println(" 发出空气与橡皮摩擦声！");}
};
class QuackNoWay implements QuackBehavior {
    public void quack( ){System.out.println(" 不能发声！");}
};
abstract class Duck{
    protected  FlyBehavior  (3) ;
    protected  QuackBehavior  (4) ;
    public  void fly( ){ (5) ;};
    public  void quack( ){ (6) ; };
    public  (7)  void display( );
};
class RubberDuck extends Duck{
    public RubberDuck( ){
        flyBehavior=new (8) ;
        quackBehavior=new (9) ;
    }
public void display( ){/*此处省略显示橡皮鸭的代码*/}
};
//其他代码省略
```

13.2 模拟试卷参考答案

13.2.1 模拟试卷一参考答案

上午科目答案与解析

(1) 【答案】C

【解析】CPU 主要由运算器、控制器、寄存器组和内部总线等部件组成。

运算器由算术逻辑单元(ALU)、累加寄存器、数据缓冲寄存器和状态条件寄存器组成，显然，ALU 不属于控制器。

控制器一般包括指令控制逻辑、时序控制逻辑、总线控制逻辑和中断控制逻辑等几个部分，有程序计数器(PC)、指令寄存器(IR)、指令译码器、地址寄存器、程序状态字寄存器(PSW)、中断机构等。

(2) **【答案】**C

【解析】由题意知，芯片数量=内存空间大小/芯片容量。

需注意的是，在计算之前要先换算为相同的单位。具体运算如下：

DFFFFH−A0000H+1=40000H，$(40000H)_{16}=2^8$K。题目中内存是按字节编址的，所以空间大小应为 2^8KB(256KB)，32K×8 比特即 32K×1 字节，所以有 256KB/32K=8。

(3)～(4) **【答案】**(3)D；(4)D

【解析】对于高速缓存 Cache，设 H_c 为其命中率，t_c 为其存取时间，t_m 为主存的访问时间，t_a 为 Cache 存储器的等效加权平均访问时间，则有 $t_a=H_ct_c+(1-H_c)t_m$。

对于本题，则：$30\times(1-H_c)+3\times H_c=3.27$，可以求得 $H_c=0.99$，即 Cache 的命中率为 99%。

对于第(4)题，主存地址为 8888888H 时，转换为二进制地址为 1000 1000 1000 1000 1000 1000 1000 B，其中块号为 10001000H，即 88H。块内地址为 1000 1000 1000 1000 1000 B，即 88888。由地址变换表知 Cache 对应的块号为 1H，所以 Cache 的地址为 188888H。

(5)～(6) **【答案】**(5)C；(6)B

【解析】指令的执行方式有串行、并行以及流水线方式，是考试的重点。

串行方式也即一条指令执行结束后下一条指令再开始执行，所以串行执行 100 条题目中的指令需要的时间为$(4\Delta t+3\Delta t+5\Delta t)\times 100=1200\Delta t$。

如果按流水线方式执行时，得：$(4\Delta t+3\Delta t+5\Delta t)+(100-1)\times 5\Delta t=507\Delta t$。

(7) **【答案】**C

【解析】本题考查网络攻击的辨别。

首先，本题的答案是非常容易给出的，向多个邮箱群发一封电子邮件有时是传递信息的需要。

网络攻击是以网络为手段窃取网络上其他计算机的资源或特权，对其安全性或可用性进行破坏的行为。再看题面，B 和 D 选项毫无疑问属于网络攻击。而 A 选项中的 ping 命令是用于网络检测的工具，ping 某台主机可测试出当前主机到某台主机的网络是否畅通。但如果有多台机器，连续不停地 ping 某台主机，则可能使主机无法响应这些数量庞大的请求，从而导致主机无法正常提供服务，这也就是 DDoS 攻击。

(8) **【答案】**D

【解析】本题考查的是网络安全的控制技术。A、B 和 C 选项都属于网络安全控制技术，下面做简单介绍。

访问控制技术：访问控制的目的是防止合法用户越权访问系统和网络资源。因此，系统要确定用户对哪些资源(比如 CPU、内存、I/O 设备程序、文件等)享有使用权以及可进行何种类型的访问操作(比如读、写、运行等)。为此，系统要赋予用户不同的权限，比如普通用户或有特殊授权的计算机终端或工作站用户、超级用户、系统管理员等，用户的权限等级是在注册时赋予的。

防火墙技术：防火墙是采用综合的网络技术设置在被保护网络和外部网络之间的一道屏障，用以分隔被保护网络与外部网络系统，防止发生不可预测的、潜在破坏性的侵入。它是不同网络或网络安全域之间信息的唯一出入口，像在两个网络之间设置了一道关卡，能根据企业的安全政策控制出入网络的信息流，

防止非法信息流入被保护的网络，并且本身具有较强的抗攻击能力。它是提供信息安全服务，实现网络和信息安全的基础设施。

入侵检测技术：是防火墙的合理补充，帮助系统对付网络攻击，扩展了系统管理员的安全管理能力(包括安全审计、监视、进攻识别和响应)，提高了信息安全基础结构的完整性。它从计算机网络系统中的若干关键点收集信息，并分析这些信息，看网络中是否有违反安全策略的行为和遭到袭击的迹象。入侵检测被认为是防火墙之后的第二道安全闸门，在不影响网络性能的情况下对网络进行监测，从而提供对内部攻击、外部攻击和误操作的实时保护。

差错控制技术：由于通信线路上总有噪声存在，所以通常情况下噪声和信息是混在一起传输的，当噪声大到一定程度时，会干扰信息，使接收到的信息出现差错。差错控制是通过一些技术手段，对接收到的信息进行正确性检查并纠正，如海明校验编码就是一种具有纠错功能的编码。

从以上分析可以得知差错控制技术不属于网络安全控制技术。

(9) **【答案】**D

【解析】本题考查路由器的作用。

路由器是连接因特网中各局域网、广域网的设备，它会根据信道的情况自动选择和设定路由，以最佳路径，按前后顺序发送信号。路由器工作于网络层，路由器的主要功能如下。

① 选择最佳的转发数据的路径，建立非常灵活的连接，均衡网络负载。

② 利用通信协议本身的流量控制功能来控制数据传输，有效地解决拥挤问题。

③ 具有判断需要转发的数据分组的功能，不仅可以根据 LAN 网络地址和协议类型，而且可以根据网间地址、主机地址、数据类型(如文件传输、远程登录或电子邮件)等，判断分组是否应该转发。对于不该转发的信息(包括错误信息)，都过滤掉，从而可以避免广播风暴，比网桥有更强的隔离作用，可以提高安全保密性能。

④ 把一个大的网络划分为若干个子网。

(10) **【答案】**B

【解析】根据《计算机软件保护条例》第二十八条规定，软件复制品的出版者、制作者不能证明其出版、制作有合法授权的，或者软件复制品的发行者、出租者不能证明其发行、出租的复制品有合法来源的，应当承担法律责任。但是根据第三十条规定，软件的复制品持有人不知道也没有合理理由应当知道该软件是侵权复制品的，不承担赔偿责任；但是，应当停止使用、销毁该侵权复制品。如果停止使用并销毁该侵权复制品将给复制品使用人造成重大损失的，复制品使用人可以在向软件著作权人支付合理费用后继续使用。

根据以上两条例应该是软件的提供者承担侵权责任，而持有者是不用承担的。

(11) **【答案】**B

【解析】根据我国专利法第九条规定，“两个以上的申请人分别就同样的发明创造申请专利的，专利权授予最先申请的人。”针对两名以上的申请人分别就同样的发明创造申请专利，专利权应授予最先申请的人。

(12) **【答案】**D

【解析】本题主要考查音频格式。

WAV 文件也称波形文件(Wave)，它来源于对声音模拟波形的采样和量化。通常使用三个参数来表示声音，量化位数、取样频率和声道数。声道有单声道和立体声之分，取样频率一般有 11 025Hz(11kHz)、

22 050Hz(22kHz)和 44 100Hz(44kHz)三种，不过尽管音质出色，但在压缩后的文件体积过大，相对其他音频格式而言是一个缺点。

乐器数字接口(Musical Instrument Digital Interface，MIDI)是 20 世纪 80 年代初为解决电声乐器之间的通信问题而提出的。MIDI 传输的不是声音信号，而是音符、控制参数等指令，它指示 MIDI 设备要做什么，怎么做，如演奏哪个音符、多大音量等。MIDI 仅仅是一个通信标准，MIDI 系统实际就是一个作曲、配器、电子模拟的演奏系统。从一个 MIDI 设备转送到另一个 MIDI 设备上去的数据就是 MIDI 信息。MIDI 数据不是数字的音频波形，而是音乐代码或称电子乐谱。由于 MIDI 文件记录的不是乐曲本身，而是一些描述乐曲演奏过程中的指令，因此它占用的存储空间比 WAV 文件小很多。

(13) **【答案】**A

【解析】人眼可见光源有两种，一种是发射光，另一种是反射光。像太阳、电灯、显示器等发出的光为发射光，而看书时，看到文字，看到图像都为反射光。如看到红色的文字，则说明该文字所用的颜料将其他颜色的光吸收掉了，而将红色的光反射出来，所以我们能看到红色的文字。因此发射光与反射光有着相反的特性。发射光利用相加混色法，反射光则以相减混色法，来进行颜色的混合。

用油墨或颜料进行混合得到的彩色称为相减混色。之所以称为相减混色，是因为减少(吸收)了人眼识别颜色所需要的反射光。

(14) **【答案】**C

【解析】本题主要考查图形的格式。

计算机中显示的图形一般可以分为两大类——矢量图和位图。

矢量图形是用一系列计算机指令来描述和记录的一幅图的内容，即通过指令描述构成一幅图的所有直线、曲线、圆、圆弧、矩形等图元的位置、维数和形状，在屏幕上显示一幅矢量图时，首先要解释这些指令，然后将描述图形图像的指令转换成屏幕上显示的形状和颜色。

位图图像是指用像素点来描述的图，即把一幅彩色图或灰度图分成许许多多的像素(点)，每个像素用若干二进制位来指定该像素的颜色、亮度和属性。位图图像在计算机内存中由一组二进制位(bit)组成，这些位定义图像中每个像素点的颜色和亮度。

由于矢量图形可通过公式计算获得，所以矢量图形文件体积一般较小。矢量图形最大的优点是无论放大、缩小或旋转等不会失真；最大的缺点是难以表现色彩层次丰富的逼真图像效果。

位图图像适合于表现比较细腻，层次较多，色彩较丰富，包含大量细节的图像，并可直接、快速地在屏幕上显示出来。但占用存储空间较大，一般需要进行数据压缩。

本题当中的地图导航系统，要求缩放而不会影响图像的质量，采用矢量图像格式最合适。

(15) **【答案】**A

【解析】结构化分析(Structured Analysis，SA)是一种面向数据流的需求分析方法，适用于分析大型数据处理系统。

(16) **【答案】**C

【解析】本题考查软件开发方法之一：原型开发方法。

原型开发方法是这样的，开发人员对用户提出的问题进行总结，就系统的主要需求取得一致意见后，开发一个原型，该原型是由开发人员与用户合作，共同确定系统的基本要求和主要功能，并在较短时间内开发的一个实验性的、简单易用的小型系统。原型应该是可以运行的，可以修改的。运行原型，反复对原型进行“补充需求-修改”这一过程，使之逐步完善，直到用户对系统满意为止。

总之，原型开发方法的核心理念是通过原型不断地获取与完善需求，最终开发出符合用户需求的软件，一般不会直接就把最终产品开发出来。

(17) **【答案】**D

【解析】本题考查软件项目的风险分析，是常考知识点。

风险分析在软件项目开发中具有重要作用，包括风险识别、风险评估、风险预测和风险控制。

风险识别：试图系统化地确定对项目估算、进度、资源分配等的威胁，常用的方法有风险识别问询法、财务报表法、流程图法、现场观察法、相关部门配合法和环境分析法等。

风险评估：对已识别的风险要进行估计和评价，风险估计的主要任务是确定风险发生的概率与后果，风险评价则是确定该风险的经济意义及处理的费/效分析，常用的方法有概率分布、外推法、多目标分析法等。

风险预测：又称风险估算，软件项目管理人员可以从影响风险的因素和风险发生后带来的损失两方面来度量风险。为了对各种风险进行估算，必须建立风险度量指标体系；必须指明各种风险带来的后果和损失；必须估算风险对软件项目及软件产品的影响；必须给出风险估算的定量结果。

风险控制：包括对风险发生的监督和对风险管理的监督，前者是对已识别的风险源进行监视和控制，后者是在项目实施过程中监督人们认真执行风险管理的组织和技术措施。

综上所述，“风险避免、风险监控和风险管理及意外事件计划”应是风险控制中需要考虑的问题。

(18) **【答案】**D

【解析】本题考查软件能力成熟度模型的概念。

软件过程能力成熟度模型将软件过程能力成熟度划分为如下五个等级。

(1) 初始级(Initial)。软件过程是无序的，有时甚至是混乱的，对过程几乎没有定义，项目的成功完全依赖个人努力和英雄式核心人物。

(2) 可重复级(Repeatable)。建立了基本的项目管理过程和实践来跟踪项目费用、进度和功能特性。制定了必要的过程纪律，能重复早先类似应用项目取得的成功。

(3) 已定义级(Defined)。已将软件管理和工程两方面的过程文档化、标准化，并综合成该组织的标准软件过程。所有的项目均使用经过批准、剪裁的标准软件过程来开发和维护软件。

(4) 已管理级(Managed)。收集对软件过程和产品质量的详细度量，对软件过程和产品都有定量的理解和控制。

(5) 优化级(Optimized)。过程的量化反馈和先进的新思想、新技术促使过程不断改进。

(19) **【答案】**A

【解析】本题考查软件开发工具的相关知识。

对应于软件开发的各种过程，软件开发工具通常有需求分析工具、设计工具、编码和排错工具、测试工具等。

软件开发工具是给系统开发人员使用的。开发工具的选择主要取决于两个因素：所开发系统的最终用户和开发人员。最终用户需求是一切软件的来源和归宿，也是影响开发工具的决定性因素；开发人员的爱好、习惯、经验也影响着开发工具的选择。严格的软件工程管理和开发人员的技术水平是软件开发成功的关键。所以在选择时，应考虑功能、易用性、稳健性、硬件要求，以及性能、服务和支持。

(20) **【答案】**B

【解析】本题考查程序设计语言的分类。

A 选项：C 语言提供了一个丰富的运算符集合和比较紧凑的语句格式，虽然 C 语言程序是由函数构成的，但它是典型的过程式(命令式)程序设计语言。

B 选项：C++、Java 和 C#都是我们十分熟悉的面向对象语言，而 Smalltalk 是第一个完全基于对象和消息概念的计算机语言。

C 选项：函数型语言是一类以λ-演算为基础的语言，其概念来自 LISP，一个在 1958 年为了人工智能应用而设计的语言。

D 选项：逻辑型语言是一类以形式逻辑为基础的语言，其代表是 PROLOG，关键操作是模式匹配。在 C/S 系统中，面向对象语言比较适合用于实现负载分散。

(21)～(22) **【答案】**(21)D；(22)A

【解析】按照用户的要求，程序设计有过程式程序设计与非过程式程序设计之分。前者是指使用过程式程序设计语言的程序设计，后者是指使用非过程式程序设计语言的程序设计。

C++语言是在 C 语言的基础上发展起来的面向对象程序设计语言，支持过程式程序设计，数据抽象程序设计、面向对象程序设计都要用到面向对象的程序设计语言。泛型(通用)程序设计就是使用模板的程序设计，也要用到高级程序设计语言。而 C 语言是典型的面向过程语言，只支持过程式程序设计。

(23) **【答案】**D

【解析】本题考查 UNIX 操作系统中设备管理的基本概念。

UNIX 的设计者们遵循一条这样的规则：UNIX 操作系统中可以使用的任何计算机资源都用一种统一的方法表示。他们选择用“文件”这个概念作为一切资源的抽象表示方法。

UNIX 系统中包括两类设备：块设备和字符设备。设备特殊文件有一个索引结点，在文件系统目录中占据一个结点，但其索引结点上的文件类型与其他文件不同，是“块”或者是“字符”特殊文件。文件系统与设备驱动程序的接口是设备开关表。硬件与驱动程序之间的接口是控制寄存器和 I/O 指令，一旦出现设备中断，根据中断矢量转到相应的中断处理程序，完成用户所要求的 I/O 任务。

(24)～(25) **【答案】**(24)D；(25)C

【解析】安全状态，是指系统能按照某种顺序如$<P_1,P_2,\cdots,P_n>$来为每个进程分配其所需资源，直至最大需求，使每个进程都能顺利完成。

先看第 24 题，首先求 T_0 时刻剩下的资源数：

$$R_1=3-(1+0+1+1)=0$$

$$R_2=5-(1+1+1+1)=1$$

$$R_3=6-(2+2+1+1)=0$$

$$R_4=8-(4+2+0+1)=1$$

可知在 T_0 时刻系统剩余的可用资源数分别为 0、1、0 和 1，且系统不再为 R_1 和 R_3 分配资源，所以不能一开始就运行需要分配 R_1 和 R_3 资源的进程。由题表可知，进程 P_2 的运行还需要分配 R_1 资源，进程 P_1 和 P_4 的运行都需要分配 R_3 资源，所以可以立即排除 A、B 和 D 选项。

现在看第 25 题的 C 选项这个安全序列 $P_3 \to P_2 \to P_1 \to P_4$，根据这个顺序系统先运行 P_3 进程，P_3 进程结束后释放它占用的资源，然后给进程 P_2 分配 P_2 需要的资源，以此类推，可知这个序列是安全序列。

(26) **【答案】**D

【解析】常见的目录结构有三种：一级目录结构、二级目录结构和多级目录结构。

一级目录结构的整个目录组织是一个线性结构，在整个系统中只需建立一张目录表，系统为每个文件分配一个目录项。其优点是简单，缺点是文件不能重名，限制了用户对文件的命名。

二级目录结构实现了文件从名字空间到外存地址空间的映射：用户名 → 文件名 → 文件内容。该结构虽然能有效地将多个用户隔离开，这种隔离在各个用户之间完全无关时是一个优点，但当多个用户之间要相互合作去共同完成一个大任务，且一个用户又需要访问其他用户的文件时，这个隔离就成为一个缺点，因为这种隔离使诸用户之间不便于共享文件。

多级目录结构像一个倒置的有限树，从树根向下，每一个结点是一个目录，叶结点是文件。采用多级目录结构的文件系统，用户要访问一个文件，必须指出文件所在的路径名，路径名是从根目录开始到该文件的通路上所有各级目录名拼起来得到的。克服了一级和二级目录的缺点，便于文件分类，可为每类文件建立一个子目录；查找速度快，因为每个目录下的文件数目较少；可以实现文件共享。但缺点是比较复杂。

综上所述，结合题目给出的选项，只有 D 选项是正确的。

(27)～(28) **【答案】**(27)B；(28)C

【解析】本题考查缺页中断和 LRU 算法的基本知识。

系统为每个作业分配三个页面的主存空间，其中一个页面用来存放程序，剩下两个页面用来存放矩阵中的数据。二维数组 A[150][100]共有 150 行 100 列，即每行 100 个整型变量。由题知每个页面可存放 150 个整型变量且矩阵 *A* 按行序存放，所以每两个页面可存放数组的三行数据，访问它们需要产生两次缺页中断。150 行总共产生 100 次缺页中断。

采用最近最少使用页面淘汰算法，每次淘汰最久未被访问的页面。因为用来存放程序的页面时时都在调用，是不会被淘汰的，所以最后留在内存中的是矩阵 *A* 的最后 3 行。

(29) **【答案】**B

【解析】RUP 中的软件生命周期在时间上被分解为四个顺序的阶段，分别是：初始阶段(Inception)、精化阶段(Elaboration)、构建阶段(Construction)和移交阶段(Transition)。

初始阶段的目标是为系统建立商业案例并确定项目的边界。

精化阶段的目标是分析问题领域，建立健全的体系结构基础，编制项目计划，淘汰项目中最高风险的元素。

在构建阶段，所有剩余的构件和应用程序功能被开发并集成为产品，所有的功能被详细测试。

移交阶段的重点是确保软件对最终用户是可用的。在发布前做产品测试，然后基于用户反馈做少量的调整。

(30)～(31) **【答案】**(30)A；(31)B

【解析】略。

(32) **【答案】**A

【解析】本题考查模块的耦合类型，通常有 7 种，如下表：

耦合类型	描　述
非直接耦合	两个模块之间没有直接关系
数据耦合	彼此之间通过数据参数来交换输入、输出信息
标记耦合	一组模块通过参数表传递记录信息
控制耦合	一个模块通过传送开关、标志、名字等控制信息，明显地控制选择另一模块的功能
外部耦合	一组模块都访问同一全局简单变量而不是同一全局数据结构，而且不是通过参数表传递该全局变量的信息

续表

耦合类型	描　述
公共耦合	都访问同一个公共数据环境
内容耦合	一个模块直接访问另一个模块的内部数据； 一个模块不通过正常入口转到另一模块内部； 两个模块有一部分程序代码重叠(只可能出现在汇编语言中)； 一个模块有多个入口

由图模块 A 和模块 E 都引用的专用数据区的内容，所以是公共耦合。

(33) **【答案】**B

【解析】本题考查白盒测试的六大覆盖方法之一——判定覆盖的相关知识。

判定覆盖又称为分支覆盖，它要求设计足够多的测试用例，使得程序中每个判定至少有一次为真值，有一次为假值，即程序中的每个分支至少执行一次。

由图，上层一个条件语句，下层两个条件语句。上层为真或假时，分别对应下层的两个判定也要各为真或假一次，共 4 次。

(34) **【答案】**C

【解析】ISO/IEC 9126 定义了描述软件质量的六个特性和相应的子特性。可维护性是与进行规定的修改所需努力有关的一组属性，包括易分析性、易改变性、稳定性、易测试性。

(35) **【答案】**D

【解析】本题考查的还是关键路径的问题。

由图可知关键路径为1→2→3→4→5→6，路径长度为 20 天。然后从终点反推，G 的最迟开始时间为 20−3=17 天，所以 E 的最迟开始时间为 17−4=13 天。

(36) **【答案】**B

【解析】本题考查软件工程基础的常识。

软件规模代码行(Line of Code，LOC)是软件规模的一种量度，它表示源代码行数。

(37) **【答案】**B

【解析】类图是最常用的 UML 图，显示出类、接口及它们之间的静态结构和关系；它用于描述系统的结构化设计。类图最基本的元素是类或者接口。

依赖关系指有两个元素 A、B，如果元素 A 的变化会引起元素 B 的变化，则称元素 B 依赖(Dependency)于元素 A。

在类中，依赖关系有多种表现形式，如一个类向另一个类发消息，则一个类是另一个类的成员，一个类是另一个类的某个操作参数等。

泛化关系(Generalization，也称概括关系)描述了一般事物与该事物中的特殊种类之间的关系，即父类与子类之间的关系。继承关系是泛化关系的反关系，也就是说子类是从父类中继承的，而父类则是子类的泛化。

关联(Association)表示两个类的实例之间存在的某种语义上的联系。例如，一个老师为某个学校工作，一个学校有多间教室。我们就认为老师和学校、学校和教室之间存在着关联关系。

聚合关系(Aggregation)是关联关系的特例。聚合关系是表示一种整体和部分的关系。如一个电话机包含一个话筒，一台电脑包含显示器、键盘和主机等就是聚合关系的例子。

(38)【答案】C

【解析】本题考查 UML 中各种对象的图形表示。

a、b、c 三种图形符号按照顺序分别表示控制对象、实体对象和边界对象。

(39)~(42) 【答案】(39)A；(40)C；(41)D；(42)B

【解析】本题主要考查 UML 中各图的意义。

用例图展现了一组用例、参与者(Actor)以及两者之间的关系。用例图通常包括用例、参与者、扩展关系、包含关系。用例图用于对系统的静态用例视图进行建模。主要支持系统的行为，即该系统在它的周边环境的语境中所提供的外部可见服务。

类图展现了一组对象、接口、协作和它们之间的关系。在面向对象系统的建模中所建立的最常见的图就是类图。

活动图专注于系统的动态视图。它对于系统的功能建模特别重要，并强调对象间的控制流程。活动图一般包括活动状态和动作状态、转换和对象。当对一个系统的动态方面进行建模时，通常有两种使用活动图的方式：对工作流建模，对操作建模。

顺序图(或称序列图)和协作图均被称为交互图，它们用于对系统的动态方面进行建模。一张交互图显示的是一个交互，由一组对象和它们之间的关系组成，包含它们之间可能传递的消息。

顺序图是强调消息时间序列的交互图，协作图则是强调接收和发送消息的对象的结构组织的交互图。

(43)【答案】A

【解析】本题考查类和实例的知识。

类是对某个对象的定义。它包含有关对象动作方式的信息，如名称、方法、属性和事件。实际上，类本身不是对象，因为它不存在于内存当中。当引用类的代码运行时，就会创建一个实例。虽然只有一个类，但可以在内存中创建多个对象。

并不是每个类都必须创建一个实例，例如接口类是不能进行实例化的。因此，答案为 A。

(44)【答案】D

【解析】本题考查几种常见的设计模式及其特点。

Adapter(适配器)设计模式的意图是将一个类的接口转换成客户希望的另外一个接口，这种模式使原本由于接口不兼容而不能一起工作的那些类可以一起工作；Iterator(迭代器)设计模式是提供一种方法顺序访问一个聚合对象中各个元素，又不需要暴露该对象的内部表示；Prototype(原型)设计模式的意图是用原型实例指定创建对象的种类，并且通过复制这些原型创建新的对象，其适用于当要实例化的类是在运行时刻指定时的情况；Observer(观察者)设计模式，定义对象间的一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象可以得到通知并且可以自动更新。因此可知答案为 D。

(45)【答案】D

【解析】面向对象分析的过程包括：从用例中提取实体对象和实体类 → 提取属性 → 提取关系 → 添加边界类 → 添加控制类 → 绘制类图 → 绘制顺序图 → 编制术语表。

依据用例描述中出现的名词和名词短语提取实体对象，必须对原始的名词和名词短语进行筛选。得到实体对象后，对实体对象进行归纳、抽象出实体类。所以名词分析是寻找实体对象的有效方法之一。

(46)【答案】A

【解析】本题主要考查设计模式的优点。模式是一种指导，在一个好的指导下，有助于问题的解决，而且会得到解决问题的最佳方法。

采用设计模式，能够复用相似问题的相同解决方案，加快设计的速度，提高一致性。

(47)【答案】C

【解析】本题考查用例图中的相关知识点。

参与者用于表示使用系统的对象，可以是一个物体或者是一个系统。而用例是用户期待系统具有的动作。

参与者可以和多个用例有关，而用例也可以和多个参与者相关，所以参与者和用例之间可以有关联的关系。

(48)【答案】B

【解析】本题考查文法到句子的推导。

S ⇒ (S,M) ⇒ ((S,M),M) ⇒ ((M,M),M) ⇒ ((MP,M),M) ⇒ ((MPP,M),M) ⇒ ((PPP,M),M) ⇒ ((fPP,M),M) ⇒ ((faP,M) ⇒ ((fac,M),M) ⇒ ((fac,MP),M) ⇒ ((fac,PP),M) ⇒ ((fac,bP),M) ⇒ ((fac,bb),M) ⇒ ((fac,bb),g)。其他选项不能由此文法推导出来。

(49)~(50)　【答案】(49)B；(50)B

【解析】由图可知，从 0 状态输入 a 到达 1 状态，从 1 输入 a 或者 b 还是回到 1 状态，同时输入 a 也可到达 2 状态。该自动机所识别的语言特点显然是 B 选项所描述的，正规式为 a(a|b)*a。

(51)【答案】B

【解析】本题还是考查的关系运算，是一个常考点。首先还是进行 R 和 S 的笛卡儿积运算，在此基础上进行选择和投影计算。

(52)~(53)　【答案】(52)B；(53)A

【解析】本题考查属性重命名机制以及基本的 SQL 语句。

52 题是将查询出来的数量的平均值重新命名为平均数量。用法为 old-name as new-name，因此选 B。

53 题要求查询出至少用了 3 家供应商的供应零件，由于可能会出现一个项目使用同一供应商的多种零件的情况，因此要加上 Distanct 加以避免。由此答案为 A。

(54)【答案】D

【解析】本题考查主码、全码和候选码等基本概念。

如果一个关系有多个候选码，那么就选定其中一个作为主码。主码的诸属性为主属性。

不包含在任何候选码中的属性为非主属性。关系模型中的所有属性组是这个关系模式的候选码，称作全码。

(55)【答案】C

【解析】本题考查元组的计算。

此表达式的含义为：从关系 *S* 中选出元组 *t*，*t* 应满足条件：对于任意的 *U* 元组都有 *t*[3]>*u*[1]，即关系 *R* 中满足 *C* 值大于 *S* 中所有 *A* 值的元组。由此经过对各选项进行比较，可得答案为 C。

(56)【答案】D

【解析】本题主要考查排他锁与共享锁的区别。

排他锁：又称写锁(eXclusive lock，简称 X 锁)，如果一个事务 *T* 对数据 *A* 加排他锁，则事务 *T* 只能读取和修改 *A*，其他事务不能对 *A* 加任何类型的锁，直到 *T* 释放为止。其作用主要就是为了避免其他事务获取资源上的锁。一般在事务的更新操作过程中始终应用排他锁。

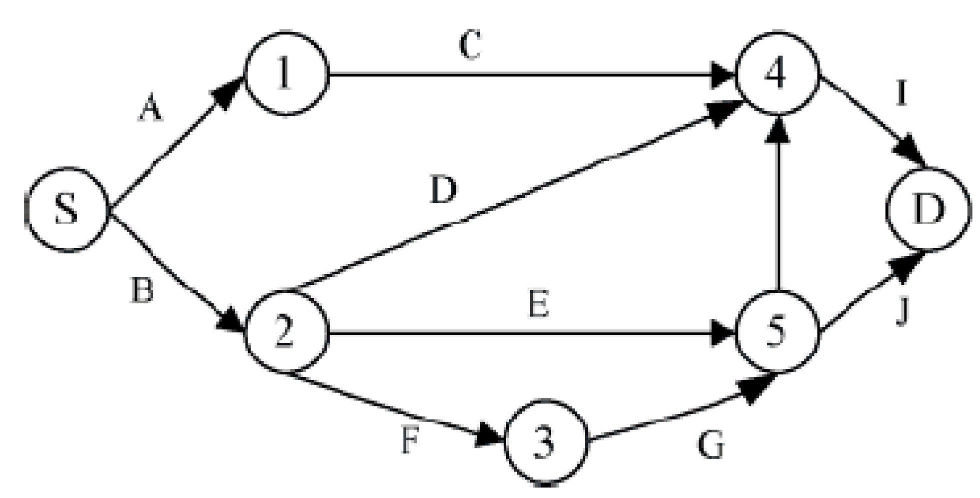
共享锁：又称读锁(Share lock，简称 S 锁)，如果事务 *T* 对数据 *A* 加共享锁，那么其他事务也只能对 *A*

加 S 锁，不能加其他锁，直到 T 释放 S 锁为止。因此，可以得出本题答案为 D。

(57) **【答案】** A

【解析】 本题考查拓扑排序的概念。

对一个有向无环图(Directed Acyclic Graph，DAG)G 进行拓扑排序，是将 G 中所有顶点排成一个线性序列，使得图中任意一对顶点 u 和 v，若$<u, v>\in E(G)$，则 u 在线性序列中出现在 v 之前。通常，这样的线性序列称为满足拓扑次序的序列，简称拓扑序列。需要注意的是：①若将图中顶点按拓扑次序排成一行，则图中所有的有向边均是从左指向右的；②若图中存在有向环，则不可能使顶点满足拓扑次序；③一个 DAG 的拓扑序列通常表示某种方案切实可行。由注意的第②点知，本题正确答案为 A。



(58)～(59) **【答案】** (58)B；(59)D

【解析】 根据排序二叉树的性质可知 58 题应该是中序。对于 59 题，在具有 n 个结点的二叉树上进行查找运算时，最坏的情况就是单支树的情况，有 n 个结点，需要比较 n 次，所以时间复杂度为 $O(n)$。

(60) **【答案】** C

【解析】 平衡二叉树，或者是一棵空树，或者是具有下列性质的二叉树：它的左子树和右子树都是平衡二叉树，且左子树和右子树的高度之差的绝对值不超过 1。

题目已经说明图示二叉树是一棵平衡二叉树，当 CR 加入一个结点并使得 CR 的高度增加 1 以后：①以 C 为顶点的子树仍为一棵平衡二叉树，因为 CL 为 $h-1$，而 CR 为 h，相差 1；②以 B 为顶点的子树仍为一棵平衡二叉树，因为 BL 为 h，而 BR(以 C 为顶点)子树的高度为 $h+1$，相差 1；③以 A 为顶点的二叉树就不平衡了，因为 AR 的高度为 h，但 AL(以 B 为顶点)子树的高度为 $h+2$ 了，相差 2。

(61) **【答案】** C

【解析】 本题考查各种排序算法的时间复杂度，在此做个总结。

排序方法	最好情况	平均时间	最坏情况
直接插入	$O(n)$	$O(n^2)$	$O(n^2)$
简单选择	$O(n^2)$	$O(n^2)$	$O(n^2)$
冒泡排序	$O(n)$	$O(n^2)$	$O(n^2)$
希尔排序	—	$O(n^{1.3})$	—
快速排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n^2)$
堆排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$
归并排序	$O(n\log_2 n)$	$O(n\log_2 n)$	$O(n\log_2 n)$
基数排序	$O(d(n+rd))$	$O(d(n+rd))$	$O(d(n+rd))$

根据题目选项和上表，显然 C 答案是正确的。

(62)～(63) **【答案】** (62)D；(63)C

【解析】 此题考的是常见内部排序算法的思想。

① 希尔排序的思想是：先取一个小于 n 的整数 d_1 作为第一个增量，把文件的全部记录分成 d_1 个组。

所有距离为 d_1 的倍数的记录放在同一个组中。先在各组内进行直接插入排序；然后，取第二个增量 $d_2<d_1$ 重复上述的分组和排序，直至所取的增量 $d_t=1(d_t<d_{t-1}<\cdots<d_2<d_1)$，即所有记录放在同一组中进行直接插入排序为止。该方法实质上是一种分组插入方法。

② 直接插入排序的思想是：每次从无序表中取出第一个元素，把它插入到有序表的合适位置，使有序表仍然有序。第一趟比较前两个数，然后把第二个数按大小插入到有序表中；第二趟把第三个数据与前两个数从后向前扫描，把第三个数按大小插入到有序表中；依次进行下去，进行了$(n-1)$趟扫描以后就完成了整个排序过程。

③ 快速排序的思想是：通过一趟排序将要排序的数据分割成独立的两部分，其中一部分的所有数据比另外一部分的所有数据都要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。

④ 堆排序的思想是(在此介绍用大根堆排序的基本思想)：①将初始文件 $R[1..n]$建成一个大根堆，此堆为初始的无序区；②将关键字最大的记录 $R[1]$(即堆顶)和无序区的最后一个记录 $R[n]$交换，由此得到新的无序区 $R[1..n-1]$和有序区 $R[n]$，且满足 $R[1..n-1].\text{keys}\leqslant R[n].\text{key}$；③由于交换后新的根 $R[1]$可能违反堆性质，故应将当前无序区 $R[1..n-1]$调整为堆，然后再次将 $R[1..n-1]$中关键字最大的记录 $R[1]$和该区间的最后一个记录 $R[n-1]$交换，由此得到新的无序区 $R[1..n-2]$和有序区 $R[n-1..n]$，且仍满足关系 $R[1..n-2].\text{keys}\leqslant R[n-1..n].\text{keys}$，同样要将 $R[1..n-2]$调整为堆。以此类推，直到无序区只有一个元素为止。

⑤ 冒泡排序的思想是：在排序过程中总是小数往前放，大数往后放，相当于气泡往上升。

题目要求得到其中第 k 个元素之前的部分排序，显然堆排序最合适，因为希尔排序、直接插入排序和快速排序都不能实现部分排序。若要把所有元素排序完成，再从结果集中把需要的数列截取出来，很明显效率远远不及堆排序。

对于第(63)题，可以从快速排序基本思想得到答案。

(64) **【答案】** A

【解析】 贪心算法是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，它所做出的仅是在某种意义上的局部最优解。而 Dijkstra 算法按照路径长度递增的次序生成从源点 s 到其他顶点的最短路径，则当前在生成的最短路径上除终点以外，其余顶点的最短路径均已生成。这是典型的贪心策略。

(65) **【答案】** B

【解析】 本题考查的是算法的时间复杂度的基本计算。

$T(n)=2T(n/2)+n$ 其实是在给 n 个元素进行快速排序时的最好情况下的时间递推关系式，其中 $T(n/2)$是一个子表需要的处理时间，n 为当次分割需要的时间。对此表达式变形得：$\frac{T(n)}{n}-\frac{T(n/2)}{n/2}=1$

用 $n/2$ 代替上式中的 n 有：$\frac{T(n/2)}{n/2}-\frac{T(n/4)}{n/4}=1$

依次替换到最后是：$\frac{T(2)}{2}-\frac{T(1)}{1}=1$

算法共需要$[\log_2 n]+1$ 次分割，将替换得到的$[\log_2 n]+1$ 个式子相加，最终得到：$\frac{T(n)}{n}-\frac{T(1)}{1}=[\log_2 n]+1$

将 $T(1)=1$ 代入得：$T(n)=n[\log_2 n]+2n$

因为 $O(n)<O(n\log n)$，而且对数的底可省略或为任意常数，所以：$T(n)=O(n\log n)=O(n\log_{10} n)=O(n\lg n)$。

(66) **【答案】** B

【解析】通过遵循 ARP(Address Resolution Protocol，地址解析协议)，只要我们知道了某台机器的 IP 地址，即可以知道其物理地址。在 TCP/IP 网络环境下，每个主机都分配了一个 32 位的 IP 地址，这种互联网地址是在网际范围标识主机的一种逻辑地址。为了让报文在物理网络上传送，必须知道对方目的主机的物理地址。这样就存在把 IP 地址变换成物理地址的地址转换问题。以以太网环境为例，为了正确地向目的主机传送报文，必须把目的主机的 32 位 IP 地址转换成为 48 位以太网的地址。这就需要在网络互联层有一组服务将 IP 地址转换为相应物理地址，这组协议就是 ARP。

(67) **【答案】**B

【解析】PING(Packet Internet Grope，因特网包探索器)，是用来检查网络是否通畅或者网络连接速度的命令。原理是这样的：网络上的机器都有唯一确定的 IP 地址，我们给目标 IP 地址发送一个数据包，对方就要返回一个同样大小的数据包，根据返回的数据包可以确定目标主机的存在，可以初步判断目标主机的操作系统等。应用格式：ping IP 地址。该命令还可以加许多参数使用，具体是输入 ping 后按 Enter 键即可看到详细说明。

Tracert(跟踪路由)是路由跟踪实用程序，用于确定 IP 数据报访问目标所采取的路径。Tracert 命令用 IP 生存时间(TTL)字段和 ICMP 错误消息来确定从一个主机到网络上其他主机的路由。其工作原理为，通过向目标发送不同 IP 生存时间值的“Internet 控制消息协议(ICMP)”回应数据包，Tracert 诊断程序确定到目标所采取的路由。要求路径上的每个路由器在转发数据包之前至少将数据包上的 TTL 递减 1。数据包上的 TTL 减为 0 时，路由器应该将“ICMP 已超时”的消息发回源系统。

ARP 是一个重要的 TCP/IP，并且用于确定对应 IP 地址的网卡物理地址。使用 ARP 命令，能够查看本地计算机或另一台计算机的 ARP 高速缓存中的当前内容。此外，使用 ARP 命令，也可以用人工方式输入静态的网卡物理/IP 地址对，你可能会使用这种方式为默认网关和本地服务器等常用主机进行这项工作，有助于减少网络上的信息量。

nslookup 是 Windows NT、2000、2003、2008 中一个监测网络中 DNS 服务器是否能正确实现域名解析的命令行工具。

(68) **【答案】**B

【解析】FTP 是 File Transfer Protocol(文件传输协议)的英文简称，而中文简称为“文传协”，用于 Internet 上的控制文件的双向传输。同时，它也是一个应用程序。用户可以通过它把自己的 PC 与世界各地所有运行 FTP 的服务器相连，访问服务器上的大量程序和信息。即其数据连接是客户端主动建立的。

(69)～(70) **【答案】**(69)D；(70)A

【解析】金融业务的特点是数据量大、数据类型多样、业务需求多样、业务需求变化快和子系统繁多。所以金融业务系统在进行网络设计时，最看重的是系统的有效性、安全性和高可用性。

利用网络技术，现代企业可以在供应商、客户、合作伙伴、员工之间实现优化的信息沟通。企业网络要求具有资源共享功能、通信服务功能、多媒体功能等，所以在进行企业网络需求分析时应着眼于企业的应用分析。

(71)～(75) **【答案】**(71)C；(72)D；(73)A；(74)B；(75)D

参考译文：原型分析强调图案系统模式的牵引作用，以记录和验证现有和/或拟议的系统。最终，该系统模型变为设计和构造一个改进的系统的蓝图。结构化分析是强调以过程为中心的技术。系统分析绘制了一系列的过程模型，称为 DFD(数据流程图)。面向对象的分析也是这样一种技术，对象集成了数据和过程。

下午科目答案与解析

试题一

【答案】

【问题 1】E1：考试委员会；E2：主讲教师；E3：学生或选课学生；E4：教务处

【问题 2】D1：学生信息文件；D2：课程单元信息文件；D3：课程信息文件；D4：课程成绩文件；D5：无效成绩文件

注：D2 和 D3 的答案可以互换。

【问题 3】起点：D4 或课程成绩文件　　终点：4 或生成成绩列表

起点：D1 或学生信息文件　　终点：5 或生成最终成绩单

起点：4 或生成成绩列表　　终点：5 或生成最终成绩单

【问题 4】程序流程图通常在进行详细设计时使用，用来描述程序的逻辑结构。

【解析】本题考查的是 DFD 的应用，属于比较传统的题目，考查点也与往年类似。

【问题 1】外部实体是在系统边界之外的个人和组织，它提供数据，或者接收数据输出。根据主要功能描述："学生成绩均由每门课程的主讲教师上传给成绩管理系统"，所以主讲教师是一个外部实体，根据顶层流程图，得知 E2 处应填"主讲教师"。"对于无效成绩，系统会单独将其保存在无效成绩文件中，并将详细情况提交给教务处"、"成绩已经被系统记录，系统会发送课程完成通知给教务处，告知该门课程的成绩已经齐全"。在这里"教务处"是一个外部实体，应填在 E4 处。"根据主讲教师核对后的成绩报告，系统生成相应的成绩列表，递交考试委员会进行审查。考试委员会在审查之后，上交一份成绩审查结果给系统。"在这里考试委员会是一个外部实体，应填在 E1 处。"对于所有通过审查的成绩，系统将生成最终的成绩单，并通知每个选课学生。"在这里"选课学生"是一个外部实体，应填在 E3 处。

【问题 2】数据存储是保存数据的地方，将来一个或者多个过程会访问这些数据。根据 0 层数据流图，在"验证学生信息"处有 3 个数据输入。根据"在记录学生成绩之前，系统需要验证这些成绩是否有效。首先，根据学生信息文件来确认该学生是否选修这门课程，若没有，那么这些成绩是无效的；如果他的确选修了这门课程，再根据课程信息文件和课程单元信息文件来验证平时成绩是否与这门课程所包含的单元相对应，如果是，那么这些成绩是有效的，否则无效"。得知由学生信息文件、课程单元信息文件、课程信息文件共三个文件帮助验证。在"生成成绩列表"处也要利用到 D1 文件，现已输入了成绩审查结果、核对后的成绩报告，只缺"学生信息文件"，所以 D1 处应填"学生信息文件"；D2 和 D3 处分别填"课程单元信息文件"和"课程信息文件"。D4 是"记录有效成绩"的一个输出文件，根据"对于有效成绩，系统将其保存在课程成绩文件中"，得知 D4 处应填"课程成绩文件"；D5 处应填"无效成绩文件"。

【问题 3】在 0 层数据流图中的"4 生成成绩列表"处没有课程成绩文件输入，不可能生成成绩列表，所以在此处缺少一条从"D4 课程成绩文件"到"4 生成成绩列表"的数据流。"5 生成最终成绩单"没有任何输入，不符合数据输入输出平衡，根据"根据主讲教师核对后的成绩报告，系统生成相应的成绩列表，递交考试委员会进行审查。考试委员会在审查之后，上交一份成绩审查结果给系统。对于所有通过审查的成绩，系统将会生成最终的成绩单"，所以应根据"学生信息文件"和"生成成绩列表"生成最终成绩单，所以还应补充从"D1 学生信息文件"到"5 生成最终成绩单"的数据流和"4 生成成绩列表"到"5 生成最终成绩单"的数据流。

【问题 4】程序流程图通常在进行详细设计时使用，用来描述程序的逻辑结构(功能需求)。

试题二

【答案】

【问题 1】(1)　1　　　　　　　(2)　*，或 n，或 m

(3)　*，或 n，或 m　　　　(4)　*，或 n，或 m

【问题 2】缺少的联系数：3

挂号单：收银员；挂号单：医师；挂号单：门诊处方

【问题 3】(5)　收银员；(6)　就诊号；(7)　药品编码，数量，单价；(8)　类型，库存，货架编号，单位，价格，单价；挂号单主键：就诊号；门诊处方主键：就诊号；处方明细主键：就诊号、药品编码

【解析】本题是一道数据库设计题，该类型的提问形式比较固定，在软件设计师考试下午题中是比较好得分的。

【问题 1】该问题是求实体间的联系，这类问题主要通过“生活常识”+“系统描述”解题。由于一名医生在不同时间段可以给多个病人看病，也就可以开多张门诊处方，而一张门诊处方由一名医生开出。所以对于医生实体与门诊处方实体之间的联系“开处方”，其联系的类型为一对多(1∶n)。所以第(1)空的答案为 1，第(2)空的答案为 n。由于一张门诊处方包含多种库存中的药品(如“××医院门诊处方单”表所示)，一种库存中的药品也可以在多张门诊处方中。所以该联系的类型为多对多(m∶n)。(3)空和(4)空均应填写 n。

【问题 2】根据“××医院门诊挂号单”可以看出，挂号单由收银员进行收费，同时收银员的编号记录到了该挂号单中，因此挂号单实体与收银员实体之间存在联系——挂号单：收银员。病人挂某个医师的号，将挂号信息记录在挂号单实体中，因此挂号单实体与医师实体之间存在联系——挂号单：医师。收银员根据挂号单和医师的手写处方生成门诊处方，所以挂号单实体与门诊处方实体之间存在联系——挂号单：门诊处方。

【问题 3】本题考查将 E-R 模型转换为关系模式。在此转换过程中，每一个实体转成一个关系模式，对于联系的转换，相对比较复杂。可单独转为关系模式，也可以将其并入实体关系模式中(注意：多对多的联系只能单独转成一个关系模式，且该关系模式的主键为各个与之关联实体主键的组合)。所以一个关系模式的属性有两类，一类是实体本身具备的属性，另一类是为了保存实体与实体之间联系而记录的属性。下面将根据实体及与之相关的联系类型结合系统说明来分析。

(1)　对于“挂号单”关系模式，由于挂号单与收银员实体有联系，且它们之间的联系没有单独转成关系模式，因此需要在“挂号单”关系模式中记录对应的收银员，因此，“挂号单”关系模式需补充属性为“收银员”。

(2)　从“××医院门诊处方单”可以得知“门诊处方”关系模式应具有的信息。但在此需要注意的是，哪些信息是“门诊处方”关系模式应直接存储的，哪些信息是可以通过查询从其他关系模式获取的。结合题目可知该关系就缺“就诊号”，若补充“就诊号”，则其他信息可通过“明细”、“收费”、“挂号门诊联系”、“开处方”等联系查询出来。

(3)　由于多张门诊处方中包含多项药品信息，而一种药品也可以属于多张门诊处方，因此通过“处方明细”关系模式来表示这种多对多的联系。并且由于每种药品的具体信息已经在“药品库存”关系模式中记录，所以，“处方明细”关系模式主要记录的是门诊处方与药品的对应关系和处方所需药品的具体数量。并且，根据题目描述，由于药品价格会发生变化，门诊管理系统必须记录处方单上药品的当前单价。因此，“药品库存”关系模式补充属性：药品编号，数量，单价。其中就诊号和药品编号一起作为主键。

(4) “药品库存”关系模式主要记录药品的详细信息和库存信息，“药品库”表中已经说明需要记录的信息，所以应补充属性：类型，库存，货架编号，单位，规格，单价。

综上所述，挂号单与门诊处方主键均为“就诊号”。而处方明细是一个多对多的联系，它的主键应为与之关联的实体主键之组合，即：(就诊号，药品编码)。

试题三

【答案】

【问题 1】A：Artist；B：Song；C：Band；D：Musician；E：Track；F：Album

【问题 2】(1)0..* (2)2..* (3)0..1 (4)1..* (5)1..* (6)1

【问题 3】类：Track　　多重度：0..1　　类：Track　　多重度：0..1

【问题 4】按任意键，选择歌曲

【解析】本道题主要是考查学生对 UML 的类图、状态图的掌握。

【问题 1】根据“每首歌曲的描述信息包括歌曲的名字、谱写这首歌曲的艺术家”和图中 A 类与 B 类之间约束为“编写”、“演奏”，所以 A 类与 B 类只能是艺术家和歌曲，又根据图上的标示的关联关系(1，0..*)，可以确定类 A 为艺术家(Artist)，类 B 为歌曲(Song)。类 B 与类 E 之间是聚集关系，根据题中“一条音轨中只包含一首歌曲或为空，一首歌曲可分布在多条音轨上”，可以得到类 E 为音轨(Track)。接下来看，类 E 与类 F 之间存在组成关系，根据“每张唱片由多条音轨构成”得到，类 F 为唱片(Album)。再来看类 C 和类 D，它们与类 A 存在“泛化”关系，根据“艺术家可能是一名歌手或一支由 2 名或 2 名以上的歌手所组成的乐队”可知，类 C 与类 D 为歌手和乐队，又因为类 C 与类 D 存在聚集关系，根据题中“一名歌手可以不属于任何乐队，也可以属于一个或多个乐队”可知，类 C 为乐队(Band)，类 D 为歌手(Musician)。

【问题 2】由第一问可知，类 C 为乐队，类 D 为歌手，题中“一支由 2 名或 2 名以上的歌手所组成的乐队。一名歌手可以不属于任何乐队，也可以属于一个或多个乐队”，则第(1)空处是 0..*，第(2)空处是 2..*。类 B 与类 E 存在聚集关系，题中“一条音轨中只包含一首歌曲或为空，一首歌曲可分布在多条音轨上”，所以第(3)空为 0..1，第(4)空为 1..*。类 E 与类 F 存在泛化关系，题中“每张唱片由多条音轨构成”，所以第(5)空为 1..*，第(6)空为 1。

【问题 3】考查的是类/对象关联中的一种特殊关联：递归关联，它描述的是同一个类的不同实例之间的关系。而类 Track 的不同实例之间恰好具有这种关系(因此对于任意一条音轨，播放器需要准确地知道，它的下一条音轨和上一条音轨是什么)。所以缺少的那条联系的两端都是类 Track，其多重度都为 0..1。下限为 0，是对应不存在上一条或下一条音轨的情况。

【问题 4】 状态图是描述系统动态行为的一种模型。这里状态图的考查仅限于能够理解它所描述的行为。状态图由状态及状态之间的迁移构成，迁移可以由相关的事件触发。问题 4 给定了两个状态“关闭”和“播放”，要求找出从“关闭”到“播放”的最短事件序列。这就要求我们能够在状态图上找到连接这两个状态的最短迁移，然后将迁移上的事件记录下来就可以了。从“关闭”状态到“播放”状态可以选择经过迁移“连接电脑”到达“联机”状态，再经过迁移“断开状态”到达状态“打开”，再从“打开”状态的初始状态“歌曲待选”，经过迁移“选择歌曲”到达“播放状态”。这样经过的事件序列为：连接电脑→电量饱和/完成复制→断开连接→选择歌曲。显然这样的事件序列远比“关闭”经过“按任意键”直接到达“打开”状态要长得多。所以从“关闭”到“播放”的最短事件序列是：按任意键，选择歌曲。

试题四

【答案】(1) root　(2) root->childptr[0]　(3) root->childptr[i]

(4) placeBoosters(p)　(5) degradation

【解析】本题考查数据结构中树的基本操作。题目说明部分对树的结构以及程序的目的有比较明确的说明。本程序的功能是在合适的位置安放信号放大器。通过对题目说明部分的分析可以得知放置信号放大器的原则是判断当前结点的 d[i]+M[i]是否大于容忍值。若大于，则在 i 处设信号放大器。如：对于结点 s，d[s]=2，M[s]=2，d[s]+M[s]=4，此时容忍值为 3，则 d[s]+M[s]>3。需要在此放置一个信号放大器。但在题目中，结点的 M 值是未提供的，所以程序应完成两个操作，第一个是求出结点的 M 值，第二个是确定当前结点是否需要加信号放大器。

下面进行具体的代码分析。

第(1)空是一个判断条件，当条件成立时，才能进入程序主体。这一空非常容易，在对树进行操作的过程中，只有当前结点不为空结点时才有必要进行相应的操作，所以此处应填 root。

通过对程序主体进行分析可知，指针 p 用于指向子结点，其初始值应为第一个子结点“childptr[0]”的指针，因此第(2)空应填 root->childptr[0]，此后 p 依次指向下一个子结点。因此第(3)空处应填 root->childptr[i]。

第(4)空是关键的一步，由于“要计算结点的 M 值，必须先算出其所有子结点的 M 值”，所以需要用到递归，利用递归来计算子结点的 M 值，故此处填 placeBoosters(p)。

第(5)空非常容易，是将已求得的 M 值，存入当前结点的 root->M 中，由于程序中计算出来的 M 记录在 degradation 中，所以此处填 degradation。

试题五

【答案】(1)ProcessRequest(aRequest)　(2)Approver *　(3)Approver::

(4)&Tammy　(5)&Meeting　(6)&Sam　(7)Larry

【解析】本题以解决某企业的采购审批分级为背景，考查考生对面向对象程序设计类的用例和继承。程序解释如下。

```
(1) class Approver {              // 审批者类
    public:
    Approver(){ successor = NULL;   }
    virtual void ProcessRequest(PurchaseRequest aRequest){
    if(successor!=NULL)
{successor->_(1) ProcessRequest (aRequest) _;}
/*Chain of Responsibility(CoR) 是用一系列类(classes)试图处理一个请求
aRequest，这些类之间是一个松散的耦合，唯一共同点是在它们之间传递 aRequest。也
就是说，来了一个请求，Director 类先处理，如果没有处理，就传递到 VicePresident
类处理，如果没有处理，就传递到 President 类处理，构成责任链。审批者类定义一个
虚函数，其子类可以重载该虚函数，用于处理采购请求，如果继任者不为空则执行相应审
批者类的处理采购请求的函数 ProcessRequest(aRequest)。*/
    }
(2) void SetSuccessor(Approver *aSuccessor){ successor = aSuccessor; }
    private:
    _(2) Approver * _ successor;
    /* 注意到此处成员变量为私有属性，只能在 Approver 类中使用。由 void SetSuccessor
    (Approver *aSuccessor){ successor = aSuccessor; }中的参数是 Approver 指针
    类型，再由 successor = aSuccessor 可知 successor 也应为 Approver 指针类型，否则无
```

```
    法赋值*/
};
(3) class Congress : public Approver {
    public:
    void ProcessRequest(PurchaseRequest aRequest){
        if(aRequest.Amount >= 500000){ /* 决定是否审批的代码省略 */   }
        else  (3) Approver::   ProcessRequest(aRequest);
        /*该类只处理 50 万元及以上的采购请求这种情况，如果是 50 万元以下，则需交给其他类
        处理。由于此处调用的是父类中的成员函数 void ProcessRequest(Purchase
        Request aRequest)，故需 Approver::ProcessRequest(aRequest)，
        此函数将请求传递给下一个继任者，即相应的子类*/
    }
};
…
(4) void main(){
   Congress Meeting;  VicePresident Sam;  Director Larry ;  President Tammy;

    // 构造责任链
    Meeting.SetSuccessor(NULL);    Sam.SetSuccessor( (4) &Tammy );
    Tammy.SetSuccessor( (5) &Meeting );Larry.SetSuccessor( (6) &Sam );
    /*由 Approver 类中的函数 void SetSuccessor(Approver *aSuccessor){ successor
    = aSuccessor; }可知这三处的参数值都是指针类型的，由于是一种责任链，当 Director
    类的对象 Larry 不能处理该请求时，必须将该请求传给上一级，即 VicePresident 类的对象
    Sam，此时需将该继任者的地址传给 Director 类的对象 Larry，否则无法传递该请求。同理
    其他上一级的对象也是如此*/

(5)PurchaseRequest aRequest;   // 构造一采购审批请求
    cin >> aRequest.Amount;   // 输入采购请求的金额
     (7) Larry  .ProcessRequest(aRequest);
    /*开始审批。一般采购请求是先由 Director 类处理，看金额是否满足，不满足再交由上一级
    的 VicePresident 类处理，不满足再如此传递下去。由上述创建的类对象 Director Larry,
    可知此处应填 Larry.ProcessRequest(aRequest)*/
    return ;
}
```

试题六

【答案】(1)ProcessRequest(aRequest)　　(2)Approver　　(3)super

(4)Tammy　　(5)Meeting　　(6)Sam　　(7)Larry

【解析】本题以用 Java 解决某企业的采购审批分级为背景，考查考生对面向对象程序设计类的用例和继承。程序解释如下。

```
if (successor != null){ successor. (1) ProcessRequest (aRequest) ;}/* Chain
of Responsibility(CoR) 是用一系列类(classes)试图处理一个请求 aRequest,这些类之间
是一个松散的耦合，唯一共同点是在它们之间传递 aRequest。也就是说，来了一个请求，Director
类先处理，如果没有处理，就传递到 VicePresident 类处理，如果还没有处理，就传递到 President
类处理，构成责任链。审批者类定义一个虚函数，其子类可以重载该虚函数，用于处理采购请求，如果继任
者不为空则执行相应审批者类的处理采购请求的函数 ProcessRequest(aRequest)*/
    }
public  void SetSuccessor(Approver aSuccesssor){ successor = aSuccesssor; }
```

```
private  (2) Approrer successor;    /* 注意到此处成员变量为私有属性，只能在
                                  Approver 类中使用。由 void SetSuccessor(Approver
                                  aSuccessor){ successor = aSuccessor; }中的参数是
                                  Approver 类型，再由 successor = aSuccessor 可知，successor
                                  也应为Approver 类型，否则无法赋值*/
};

class Congress extends Approver {
public  void ProcessRequest(PurchaseRequest aRequest){
        if(aRequest.Amount >= 500000){ /* 决定是否审批的代码省略 */    }
else  (3) super  .ProcessRequest(aRequest);  /*该类只处理 50 万元及以上的采购请
                        求这种情况，如果是 50 万元以下，则需交给其他类处理。由于此处调用的是父
                        类中的成员函数 void ProcessRequest(PurchaseRequest aRequest),
                        故需 super.ProcessRequest(aRequest)，此函数将请求传递给下一个继
                        任者，即相应的子类*/
    }
};
…
public class rs {
    public static void main(String[] args) throws IOException {
        Congress Meeting = new Congress();
        VicePresident Sam = new VicePresident();
        Director Larry = new Director();
        President Tammy = new President();
        // 构造责任链
        Meeting.SetSuccessor(null);
        Sam.SetSuccessor( (4) Tammy  );
        Tammy.SetSuccessor( (5) Meeting  );
        Larry.SetSuccessor( (6) Sam  );
/*由 Approver 类中的函数 void SetSuccessor(Approver aSuccessor){ successor =
aSuccessor; }可知，这三处的参数值都是 Approver 或者其子类类型的对象，由于是一种责任链，
当 Director 类的对象 Larry 不能处理该请求时，必须将该请求传给上一级，即 VicePresident
类的对象 Sam，此时需将该继任者的地址传给 Director 类的对象 Larry，否则无法传递该请求。
同理，其他上一级的对象也是如此*/
// 构造一采购审批请求
        PurchaseRequest aRequest = new PurchaseRequest();
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        aRequest.Amount = Double.parseDouble(br.readLine());
         (7) Larry  .ProcessRequest(aRequest);
        /* 开始审批。一般采购请求是先由 Director 类处理，看金额是否满足，不满足再交由上一级的
VicePresident 类处理，不满足再如此传递下去。由上述创建的类对象 Director Larry,可知
此处应填 Larry.ProcessRequest(aRequest)*/
```

13.2.2 模拟试卷二参考答案

上午科目答案与解析

(1) 【答案】D

【解析】本题考查两个带符号数的减法。

两个带符号数算术运算的溢出可根据运算结果的符号位 SF 和进位标志 CF 判别。该方法适用于两同号数求和或异号数求差时判断溢出。溢出的逻辑表达式为：VF=SF⊕CF。即 SF 和 CF 异或结果为 1 时表示发生溢出，异或结果为 0 时则表示没有溢出。

(2) 【答案】B

【解析】本题考查 Cache 的设置目的，它是介于 CPU 和主存之间的小容量存储器。

由于 CPU 的速度比主存的读取速度快得多，为解决这种不匹配，在它们之间设置高速缓冲存储器 Cache，将主存中的内容事先调入 Cache 中，CPU 直接访问 Cache 的时间短得多，这样就大大提高了 CPU 对主存的访问效率，同时也提高了整个计算机系统的效率。

(3) 【答案】C

【解析】给定起止地址码的内存容量=终止地址−起始地址+1，所以计算过程如下：

43FFH−4000H+1=400H，$(400H)_{16}=2^{10}$，因此有：$2^{10}/4=2^{8}=256$。

(4)～(5) 【答案】(4)B；(5)A

【解析】本题考查操作数的几种基本寻址方式。操作数的寻址方法在此考点的第三个重点中讲得比较清楚了，在此不再赘述。

其中立即寻址方式的特点是指令执行时间很短，因为不需要访问内存来取操作数。

(6) 【答案】B

【解析】在指令流水线上，执行周期取决于时间最长的子过程，由题目知该流水线的瓶颈为第三步，即第三步所用时间最长。

设第三步的时间为 $m\Delta t$，则由 2009 年 5 月真题 6 解析有：$(1+1+m+1)\Delta t+m(50-1)\Delta t=153\Delta t$，解方程得 $m=3$。

(7) 【答案】C

【解析】本题考查网络攻击的相关知识，常见的网络攻击方式如下。

中间人攻击(Man-in-the-Middle Attack，MITM 攻击)是一种“间接”的入侵攻击，这种攻击模式是通过各种技术手段将受入侵者控制的一台计算机虚拟放置在网络连接中的两台通信计算机之间，这台计算机就称为“中间人”。然后入侵者用这台计算机模拟一台或两台原始计算机，使“中间人”能够与原始计算机建立活动连接并允许其读取或篡改传递的信息，然而两个原始计算机用户却认为他们是在互相通信，因而这种攻击方式并不很容易被发现。所以中间人攻击很早就成了黑客常用的一种古老的攻击手段，并且一直到今天还具有极大的扩展空间。

在网络安全方面，MITM 攻击的使用是很广泛的，曾经猖獗一时的 SMB 会话劫持、DNS 欺骗等技术都是典型的 MITM 攻击手段。如今，在黑客技术越来越多地运用于以获取经济利益为目标的情况下，MITM 攻击成为对网银、网游、网上交易等最有威胁并且最具破坏性的一种攻击方式。

DoS 的攻击方式有很多种，最基本的 DoS 攻击就是利用合理的服务请求来占用过多的服务资源，从而使服务器无法处理合法用户的指令。DDoS 攻击手段是在传统的 DoS 攻击基础之上产生的一类攻击方式。单一的 DoS 攻击一般是采用一对一方式的，当被攻击目标 CPU 速度低、内存小或者网络带宽小等各项性能指标不高，它的效果是明显的。随着计算机与网络技术的发展，计算机的处理能力迅速增长，内存大大增加，同时也出现了千兆级别的网络，这使得 DoS 攻击的困难程度加大了——目标对恶意攻击包的“消化能力”加强了不少。例如如果攻击软件每秒钟可以发送 3000 个攻击包，而用户的主机与网络带宽每秒钟

可以处理 10 000 个攻击包，这样一来攻击就不会产生什么效果。

MAC/CAM 攻击：交换机主动学习客户端的 MAC 地址，并建立和维护端口与 MAC 地址的对应表以此建立交换路径，这个表就是我们通常所说的 CAM 表。CAM 表的大小是固定的，不同的交换机的 CAM 表大小不同。MAC/CAM 攻击是指利用工具产生欺骗 MAC，快速填满 CAM 表，交换机 CAM 表被填满后，交换机以广播方式处理通过交换机的报文，这时攻击者可以利用各种嗅探攻击获取网络信息。CAM 表满了以后，流量以洪泛方式发送到所有接口，也就代表 TRUNK 接口上的流量也会发给所有接口和邻接交换机，会造成交换机负载过大、网络缓慢和丢包，甚至瘫痪。

(8) **【答案】**A

【解析】本题考查路由汇聚算法。

路由汇聚的“含义”是把一组路由汇聚为一个单个的路由广播。它的最终结果和最明显的好处是缩小了网络上的路由表的尺寸。需要注意地址覆盖，即网络号部分是相同的。

由题目地址和选项地址，则有：

202.118.133.0/24　11001110.01110110.10000101

202.118.130.0/24　11001110.01110110.10000010

202.118.128.0/21　11001110.01110110.10000000

202.118.128.0/22　11001110.01110110.10000000

202.118.130.0/22　11001110.01110110.10000010

202.118.132.0/20　11001110.01110110.10000100

仔细观察便可知，题目中给出的 4 个地址只有前 21 位是相同的，所以 A 的地址是能够覆盖的。

(9) **【答案】**B

【解析】Tracert(跟踪路由)是路由跟踪实用程序，用于确定 IP 数据访问目标所采取的路径。因此，通过该命令可以查看在哪段路由出现问题。

(10) **【答案】**B

【解析】著作权、邻接权、专利权、商标权、商业秘密权和集成电路分布图设计属于知识产权的范围。物权不属于知识产权的范围。

(11) **【答案】**B

【解析】题目中非常明显地指出了“某开发人员不顾企业有关保守商业秘密的要求”，所以其行为侵犯了企业商业秘密权。商业秘密的概念如下。

商业秘密是指不为公众所知，具有经济利益，具有实用性，并且已经采取了保密措施的技术信息与经营信息。在《反不正当竞争法》中对商业秘密进行了保护，以下均为侵犯商业秘密的行为：以盗窃、利诱、胁迫等不正当手段获取别人的商业秘密；披露、使用不正当手段获取的商业秘密；违反有关保守商业秘密要求的约定，披露、使用其掌握的商业秘密。

(12) **【答案】**B

【解析】W3C 是英文 World Wide Web Consortium 的缩写，中文意思是 W3C 理事会或万维网联盟。W3C 组织是制定网络标准的一个非营利组织，像 HTML、XHTML、CSS、XML 的标准就是由 W3C 来定制。

XML(Extensible Markup Language)即可扩展标记语言，它与 HTML 一样，都是 SGML(Standard Generalized Markup Language，标准通用标记语言)。

SMIL 是同步多媒体集成语言(Synchronized Multimedia Integration Language)的缩写，念做 smile。它是由 W3C 组织规定的多媒体操纵语言。

VRML(Virtual Reality Modeling Language，虚拟现实建模语言)是一种用于建立真实世界的场景模型或人们虚构的三维世界的场景建模语言，也具有平台无关性，是目前 Internet 上基于 WWW 的三维互动网站制作的主流语言。

(13) **【答案】**D

【解析】点距是指屏幕上相邻两个同色像素单元之间的距离，即两个红色(或绿、蓝)像素单元之间的距离。

显示器上显示的文本和图像都是由像素点组成的，像素点越密(即像素点距越小)，越不容易看出其中的间隙，这样显示出来的图像越清晰。市场上常见点距为 0.31mm、0.28mm、0.26mm 的显示器。对于本题应选点距最小的 0.28。

(14) **【答案】**C

【解析】 2^{16}=65 536。

(15) **【答案】**D

【解析】本题考查软件工程中的版本控制工具。

CVS(Concurrent Versions System)是一种广泛应用的、开源的、透明于网络的版本控制系统，是用于开发人员协作开发时保持版本一致的软件或标准。它只保存一份源码并记录所有对它的改动。当开发者需要文件的某个特定版本时，CVS 会根据那些记录重建出需要的版本。

(16) **【答案】**C

【解析】软件生存周期包括系统分析、软件项目计划、需求分析、设计(概要设计和详细设计)、编码、测试和维护。其中编码阶段只有高级程序员和程序员参与，无须用户参与。

(17) **【答案】**B

【解析】略。

(18)～(19) **【答案】**(18)D；(19)A

【解析】本题考查软件开发项目管理中的进度管理技术。

甘特图：可以直观地表明任务计划在什么时候进行，以及实际进展与计划要求的对比。管理者由此可以非常方便地弄清每一项任务(项目)还剩下哪些工作要做，并可评估工作是提前还是滞后，抑或正常进行。

PERT(性能评审技术)图：是一个项目管理工具，用于规划、组织和调整项目内的任务。一个 PERT 图显示了一个项目的图形解释，这种图是网络状的，由号码标记的结点组成，结点由带标签的带方向箭头的线段连接，展现项目中的事件或转折点，以及展现项目中的任务。

PERT/CPM 图：是一个项目管理工具，用于规划、组织和调整项目内的任务。PERT 是基于性能评审技术，一种美国海军于 20 世纪 50 年代发展起来的管理潜艇导弹计划的方法。另外一种是关键途径方法(CPM)，它是在同样的时间内由私营部门发展的项目管理办法，现在已成为 PERT 的同义词。

鱼骨图：因其图形像鱼骨而得名。就是将造成某项结果的众多原因，以系统的方式图解之，也就是以图表的方式来表达结果与原因的关系，不能用来描述项目开发的进度安排。

(20) **【答案】**A

【解析】本题考查编译型语言、解释型语言和脚本语言的基本概念的判断。编译型语言：用该语言编写的程序执行前需要编译器将源程序翻译为目标代码程序，然后在目标机器上运行代码程序。C 语言就是

这种语言。解释型语言：用该语言编写的程序无须编译为目标代码即可执行。脚本语言：是为了缩短传统的编写-编译-链接-运行(edit-compile-link-run)过程而创建的计算机编程语言，通常是解释运行而非编译运行。

(21) **【答案】**C

【解析】本题考查源程序的错误类型。源程序的错误类型有两种，分别是语法错误和语义错误(逻辑错误)。

对于语法错误，往往编译系统比较容易诊断出来，常见的语法错误有非法字符、拼写错误、缺少分号、该匹配的关键字不匹配等。对于语义错误，编译系统是很难诊断出来的，也就是说程序运行时编译系统不报错，但是运行结果却不正确。常见的语义错误有类型不一致、参数不匹配、死循环、作为除数的变量为 0 等。

(22) **【答案】**D

【解析】正规式只能表示给定结构的固定次数的重复或者没有指定次数的重复。本题中指定了 m 的重复次数，但是 m 是不固定的，所以，不能用正规式表示(A、C 错误)。

对于每个非确定的有限自动机，都有一个与其等价的正规式，因此 B 不正确。

上下文无关文法的描述功能比正规式更强大，可以表示次数不固定的重复，所以 D 是正确的。

(23)~(24) **【答案】**(23)A；(24)B

【解析】本题考查设备驱动程序的基本概念和主要任务。

设备驱动程序是一种可以使计算机和设备通信的特殊程序，可以说相当于硬件的接口，操作系统只能通过这个接口来控制硬件设备的工作，假如某设备的驱动程序未能正确安装，便不能正常工作。

正因为这个原因，驱动程序在系统中所占的地位十分重要，一般当操作系统安装完毕后，首要的便是安装硬件设备的驱动程序。不过，大多数情况下，并不需要安装所有硬件设备的驱动程序，例如硬盘、显示器、光驱、键盘、鼠标等就不需要安装驱动程序，而显卡、声卡、扫描仪、摄像头、Modem 等就需要安装驱动程序。

第 24 题考查驱动程序的任务。驱动程序的作用是将硬件本身的功能告诉操作系统，然后完成硬件设备电子信号与操作系统及软件的高级编程语言之间的互相翻译。当操作系统需要使用某个硬件时，工作顺序为上层软件→操作系统→驱动程序→硬件，由此可知，驱动程序在操作系统和硬件之间工作，与上层软件没有关系。所以 B 答案正确。

(25) **【答案】**B

【解析】8KB=2^{13}，所以页内地址有 13 位。逻辑地址 9621 转换为二进制为 10 0101 1000 1100，最高一位为页号，低 13 位为页内偏移量，所以逻辑地址 9621 的页号为 1，由图可知其物理块号为 3，转换为二进制是 11。最后把物理块号和页内偏移地址拼合得 110 0101 1000 1100，即为十进制的 25 996。

(26)~(27) **【答案】**(26)C；(27)B

【解析】本题考查对“进程三态模型”的理解。

其中：1 表示就绪进程被调度；2 表示运行进程的时间片到了；3 表示运行进程执行了 P 操作，进程进入阻塞状态；4 表示被阻塞进程等待的事件发生了。

再看第 27 题，当一个正在运行的进程时间片到了以后，该进程将从运行态转换为就绪态(原因 2)，同时，需要调入另外一个处于就绪态的进程，使之转换为运行态(原因 1)。所以答案是 2→1。

(28) **【答案】**B

【解析】SPOOLing (Simultaneous Peripheral Operation On-Line，外部设备联机并行操作)系统主要包括以下三部分。

① 输入井和输出井：这是在磁盘上开辟出来的两个存储区域。输入井模拟脱机输入时的磁盘，用于收容 I/O 设备输入的数据。输出井模拟脱机输入时的磁盘，用于收容用户程序的输出数据。

② 输入缓冲区和输出缓冲区：这是在内存中开辟的两个缓冲区。输入缓冲区用于暂存由输入设备送来的数据，以后再传送到输出井。输出缓冲区用于暂存从输出井送来的数据，以后再传送到输出设备。

③ 输入进程和输出进程：输入进程模拟脱机输入时的外围控制机，将用户要求的数据由输入设备送到输入缓冲区，再送到输入井。当 CPU 需要输入设备时，直接从输入井读入内存。输出进程模拟脱机输出时的外围控制机，把用户要求输入的数据，先从内存送到输出井，待输出设备空闲时，再将输出井中的数据，经过输出缓冲区送到输出设备上。

综上可知，SPOOLing 技术是利用磁盘提供虚拟设备，B 答案正确。

(29) **【答案】** D

【解析】 略。

(30) **【答案】** B

【解析】 本题考查白盒测试的六大覆盖方法之一——语句覆盖的相关知识。

语句覆盖是指选择足够的测试用例，使得运行这些测试用例时，被测程序的每一个语句至少执行一次。图中有两个判断语句 X>0 和 Y>0，但是是顺序执行的，且题目求至少需要的测试案例，所以选择 2 个测试案例就可以了。

(31) **【答案】** B

【解析】 耦合是指模块之间联系的紧密程度，耦合度越高则模块的独立性越差；内聚是指模块内部各元素之间联系的紧密程度，内聚度越低模块的独立性越差。所以设计软件时，应力求高内聚低耦合。

(32) **【答案】** A

【解析】 统一过程(UP)的基本特征是“用例驱动、以架构为中心的和受控的迭代式增量开发”。其核心的工作是流，包括捕获用户需求、分析、设计、实现和测试等。

工作流程如下：①开发人员通过和用户的沟通、了解，捕获用户需求并制作软件用例图，从而得到软件的用例模型；②分析并设计满足这些用例的系统，得到分析模型、设计模型和实施模型，进而实现该系统；③通过测试模型来验证系统是否满足用例中描述的功能。

由上可知，需求捕获过程中得到的用例将各个核心工作流结合为一个整体，驱动整个软件开发过程。

(33) **【答案】** B

【解析】 本题考查黑盒测试的等价类划分技术。

等价类是指某个输入域的子集合。在该子集合中，各个输入数据对于揭露程序中的错误都是等效的，并合理地假定：测试某等价类的代表值就等于对这一类其他值的测试，因此，可以把全部输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件就可以用少量代表性的测试数据取得较好的测试结果。等价类划分可有两种不同的情况：有效等价类和无效等价类。

① 有效等价类：是指对于程序的规格说明来说是合理的、有意义的输入数据构成的集合。利用有效等价类可检验程序是否实现了规格说明中所规定的功能和性能。

② 无效等价类：与有效等价类的定义恰好相反，指对程序的规格说明是不合理的或无意义的输入数据所构成的集合。对于具体的问题，无效等价类至少应有一个，也可能有多个。

对于本题，输入范围为 16～40，所以这个区域内的为有效等价类，小于 16 或大于 40 的为无效等价类，即一个有效等价类，两个无效等价类。

(34) **【答案】** C

【解析】 软件可靠性(Software Reliability)是软件系统的固有特性之一，它表明了一个软件系统按照用户的要求和设计的目标，执行其功能的正确程度。软件可靠性与软件缺陷有关，也与系统输入和系统使用有关。理论上讲，可靠的软件系统应该是正确、完整、一致和健壮的。在四个选项中，A、B、D 都不属于软件可靠性。

(35) **【答案】** C

【解析】 本题考查 FTR 指导原则，包括如下几个方面：①软件评审是评审软件产品，不要涉及对软件生产者能力的评价；②评审前要制订严格的评审计划，并严格遵守预计的日程安排；③对评审中出现的问题要记录在案，不要过多地讨论解决方案，把问题留给软件生产者来解决；④要限制参与者人数，并要求参加评审的人员在评审会之前仔细阅读文档，做好充分的准备。

(36) **【答案】** D

【解析】 关键路径上的活动为关键活动。由图可知，本题的关键路径有两条：1.S → 2 → 5 → 4 → D；2.S → 2 → 5 → D。路径的长度均为 20。

(37) **【答案】** A

【解析】 边界类描述的是系统外部环境和系统内部运作之间的交互，它工作在外部环境和系统之间，边界对象表示一个交互窗口。实体类是存储和管理系统内部的信息，它可以有行为，但必须和它所代表的对象密切相关，实体类是独立于系统外部环境的。控制类主要描述特定的 Use Case 的控制行为，与特定的 Use Case 实现密切相关，可以有效地降低边界类和实体类之间的耦合，使系统对于外部环境的变化能更好适应。因此本题答案为 A。

(38)～(41) **【答案】** (38)C；(39)A；(40)B；(41)D

【解析】 本题考查用例图的相关知识。

用例图通常包括用例(Use Case)、参与者(Actor)、系统边界和箭头。用例图用于对系统的静态用例视图进行建模。主要支持系统的行为，即该系统在它的周边环境的语境中所提供的外部可见服务。

本题中的 X1、X2、X3 表示参与者，椭圆表示用例，小人表示参与者。

用例图中包含泛化关系、包含关系、扩展关系三种关系。

泛化关系是一种一般-特殊关系，利用这种关系，子类可以共享父类的结构和行为。UC3 是抽象用例，具体功能由 UC1 和 UC4 实现，因此 X1 可通过 UC4 和 UC1 用例与系统进行交互。

包含关系把几个用例的公共步骤分离成一个单独的被包含用例。用例间的包含关系允许将提供者用例的行为包含到用户的用例事件中，把包含用例称为客户用例，被包含用例称为提供者用例，包含用例给客户用例提供功能。由题图可知，UC4 与 UC5 之间是包含关系，UC5 是 UC4 的必需部分。

扩展关系是把新行为插入到已有的用例中的方法。基础用例提供一组扩展点，在这些扩展点可以添加新的行为，而扩展用例提供了一组插入片段，这些片段能够被插入到基础用例的扩展点。由题图知，UC2 与 UC4 之间是扩展关系，UC2 是 UC4 的可选部分。

(42) **【答案】** B

【解析】 本题考查 UML 中各种图的功能。

类图展现了一组对象、接口、协作和它们之间的关系。在面向对象系统的建模中所建立的最常见的图

就是类图。

用例图展现了一组用例、参与者(Actor)以及两者之间的关系。用例图通常包括用例、参与者、扩展关系、包含关系。用例图用于对系统的静态用例视图进行建模。主要支持系统的行为，即该系统在它的周边环境的语境中所提供的外部可见服务。

对象图展现一组对象和它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图相同，这些图给出了系统的静态设计视图或静态进程视图，但是它们是从真实的或原型案例的角度建立的。

协作图主要强调收发信息的对象的结构组织，序列图和协作图都是交互图。交互图展示了一种交互，它由一组对象和它们之间的关系组成，包括它们之间可能发送的消息。序列图和协作图是同构的，它们可以相互转换。

(43) **【答案】**C

【解析】面向对象分析的目的是获得对应用问题的理解，理解的目的是确定系统的功能、性能要求。

面向对象分析阶段包含 5 个活动：认定对象、组织对象、描述对象间的相互作用、定义对象的操作、定义对象的内部信息。分析阶段最重要的是理解问题域的概念，其结果将影响整个工作。A、B、D 都属于对象分析阶段，而 C 答案确定接口规格是面向对象设计。

(44)～(45)　**【答案】**(44)A；(45) B

【解析】本题考查 MVC 模式。

MVC 模式是一个复杂的架构模式，其实现起来也是非常复杂的一个过程。

视图代表用户交互的界面，对于 Web 应用来说，可以概括为 HTML 界面，也有可能是 XHTML、XML 和 Applet。

模型是业务流程/状态的处理以及业务规则的一些规定。业务模型的设计是 MVC 的主要核心。

控制可以理解为用户接收请求，将模型和视图匹配在一起，共同完成用户的请求。

(46)～(47)　**【答案】**(46)B；(47)A

【解析】本题考查 UML 中的基本概念。UML 中有 4 种关系：依赖、关联、泛化和实现。

泛化关系(Generalization，也称概括关系)描述了一般事物与该事物中的特殊种类之间的关系，即父类与子类之间的关系。

关联(Association)表示两个类的实例之间存在的某种语义上的联系。

聚合关系(Aggregation)是关联关系的特例。聚合关系是表示一种整体和部分的关系。

依赖：对于两个相对独立的对象，当一个对象负责构造另一个对象的实例，或者依赖另一个对象的服务时，这两个对象之间主要表现为依赖关系。

(48)　**【答案】**C

【解析】题目中已经说明序言性注释是辅助理解程序的，往往是对程序的整体说明。嵌入在程序中的 SQL 语句是通过一些应用程序接口嵌套在程序中的，属于程序的一部分，不属于注释。

(49) **【答案】**D

【解析】本题考查 while 循环和 do-while 循环的基本知识，这两个循环结构在程序中是非常常见的，只要搞清楚其执行顺序，结果不难判断。

while 循环先进行条件判断，如条件成立则执行循环体，否则退出循环，可知，条件不成立的那一次判断是没有执行循环体的，所以循环体的执行次数要比判断次数少 1。

do-while 循环先执行循环体，再进行条件判断，可知，在没有进行判断以前就已经执行了 1 次循环体，所以进行判断的次数和执行循环体的次数刚好相等。

(50) **【答案】** D

【解析】 $1^*(0|01)^* \Leftrightarrow 1^*0^* \& 1^*(01)^*$，可知得不出长度的奇偶性(A、C 错误)，也得不到开始和结尾字符都为 1 的串(B 错误)。

(51) **【答案】** A

【解析】 本题主要考查数据库管理系统中的安全控制机制。

首先我们来看数据库的完整性约束。数据库完整性(Database Integrity)是指数据库中数据的正确性和相容性。数据库完整性由各种各样的完整性约束来保证，因此数据库完整性设计就是数据库完整性约束的设计。这跟数据库的安全性并无关联。所以排除选项 A。再从备选答案来看，C 与 D 非常明显属于安全控制机制。

然后我们着重说明“视图”。视图是一个虚拟表，并不真实存在。其内容由查询定义。同真实的表一样，视图包含一系列带有名称的列和行数据。但是视图并不在数据库中以存储的数据值集形式存在。行和列数据来自由定义视图的查询所引用的表，并且在引用视图时动态生成。通过视图，可以使各种数据库用户只能访问其具备权限的数据，这样提高了数据的安全性。所以也属于安全机制的一种。据此答案为 A。

(52)～(53) **【答案】** (52)A；(53)D

【解析】 本题主要考查传递依赖。

传递依赖：如果 X→Y，Y→A，且 Y 不依赖 X，A 不是 Y 的子集，那么称 X→A 是传递函数依赖。A 答案正确。B 答案满足函数依赖的合并规则；C 答案满足函数依赖的增广率；D 答案满足引理规则。

第 53 空，很明显是 D 答案错误，因为 A、B 依赖于 C，不可能得到 A、B 都完全依赖于 C。

(54) **【答案】** B

【解析】 本题主要考查的知识点是数据库的几个范式之间的区别。

首先表 S 中每个属性都是不可再分的，因此符合 1NF 的要求；

然后每一非主属性完全依赖于主属性 Sno，满足 2NF；

而 Zip→City 是传递依赖，不满足 3NF。因此表 S 最高满足 2NF。

(55)～(56) **【答案】** (55)A；(56)D

【解析】 本题考查数据库关系运算。

$\bowtie$ 是连接运算符，它是由两个关系的笛卡儿积中选取属性间满足一定条件的元组。构成笛卡儿积的要求：首先进行投影找到共同的元素，然后再通过选择不同属性的列。因此第 55 题答案为 A。经过计算，可知第 56 题答案为 D。

(57)～(58) **【答案】** (57)C；(58)B

【解析】 本题考查栈和队列的插入和删除操作特点。

栈的操作特点是后进先出，而队列是先进先出。所以按照题中给的已知条件可知队列的出队序列也即栈的出栈序列：c、d、b、a、e。求栈的容量，须知栈底元素出栈前栈中元素最多时是几个。根据入栈序列 a、b、c、d、e 和出栈序列 c、d、b、a、e，不难看出栈容量至少为 3 个。

(59) **【答案】** D

【解析】 题目中已经把受限双队列的操作特性说清楚了。

A 选项：元素 8、1、4、2 依次进入队列，此时，元素 2 先出队列，元素 8、1、4 再依次出队，可得到

输出序列 2、8、1、4。

B 选项：元素 8、1 先进入队列，然后元素 1 出队，元素 4 入队并出队，元素 2 入队并出队，最后元素 1 出队，得到输出序列 1、4、8、2。

C 选项：元素 8、1、4 依次进入队列，然后元素 4 出队，元素 2 入队并出队，最后元素 1 和 8 依次出队，得到输出序列 4、2、1、8。

D 序列是得不到的。

(60) **【答案】** C

【解析】 线性数据结构有顺序存储结构和链式存储结构，其特点分别如下。

顺序存储结构是把逻辑上相邻的结点存储在物理位置相邻的存储单元中，结点间的逻辑关系由存储单元的邻接关系来体现。

链式存储结构不要求逻辑上相邻的结点在物理位置上亦相邻，结点间的逻辑关系是由附加的指针字段表示的。

因此，链式存储结构在插入或删除元素时就显得非常方便，因为不需要移动其他数据，由指针指示位置即可。

(61)～(62)　**【答案】** (61)A；(62)B

【解析】 哈夫曼树又称最优二叉树，是一种带权路径长度最短的二叉树。所谓树的带权路径长度，就是树中所有的叶结点的权值乘上其到根结点的路径长度(若根结点为 0 层，叶结点到根结点的路径长度为叶结点的层数)。

哈夫曼算法如下。

① 对给定的 n 个权值$\{W_1,W_2,W_3,\cdots,W_i,\cdots,W_n\}$构成 n 棵二叉树的初始集合 $F=\{T_1,T_2,T_3,\cdots,T_i,\cdots,T_n\}$，其中每棵二叉树 T_i 中只有一个权值为 W_i 的根结点，它的左右子树均为空。(为方便在计算机上实现算法，一般还要求以 T_i 的权值 W_i 的升序排列)

② 在 F 中选取两棵根结点权值最小的树作为新构造的二叉树的左右子树，新二叉树的根结点的权值为其左右子树的根结点的权值之和。

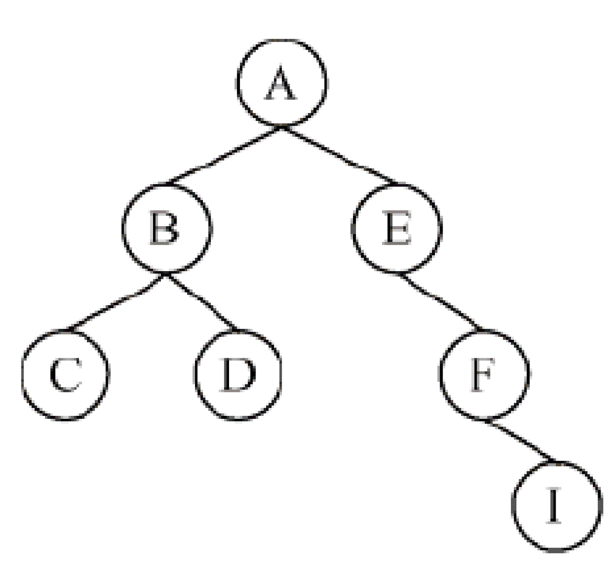
③ 从 F 中删除这两棵树，并把这棵新的二叉树同样以升序排列加入到集合 F 中。

④ 重复(2)和(3)两步，直到集合 F 中只有一棵二叉树为止。

由上述步骤构造出来的哈夫曼树是选项 A，带权路径长度为$(12+6)\times3+(15+23+29)\times2=188$。

(63) **【答案】** C

【解析】 题目中已经给出了中序序列 CBDAEFI 和先序序列 ABCDEFI，要想求二叉树的高度，最直观的方法就是构造一棵二叉树，如下图所示。



显然，该二叉树的高度为 4。

(64)　**【答案】** A

【解析】 由题可知，这类问题通常是把一个复杂的问题层层转化为一个规模较小的问题来求解。迭代

算法是用计算机解决问题的一种基本方法。它利用计算机运算速度快、适合做重复性操作的特点，让计算机对一组指令(或一定步骤)进行重复执行，在每次执行这组指令(或这些步骤)时，都从变量的原值推出它的一个新值。

递归算法是一种直接或者间接地调用自身的算法。在计算机编写程序中，递归算法对解决一大类问题是十分有效的，它往往使算法的描述简洁而且易于理解。递归算法的实质是把问题转化为规模缩小了的同类问题的子问题，然后递归调用函数(或过程)来表示问题的解。

两种算法比较，显然迭代方法具有更高的时空效率。

(65)【答案】B

【解析】贪心算法是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，它所做出的仅是在某种意义上的局部最优解。这个找零钱的方法使用的就是这种思想。

(66)【答案】D

【解析】本题考查 IP 地址的分类。

① A 类 IP 地址。由 1B 的网络地址和 3B 的主机地址组成，网络地址的最高位必须是 0，地址范围 1.0.0.1～126.255.255.254 可用的 A 类网络有 126 个，每个网络能容纳 1 677 214 个主机。

② B 类 IP 地址。由 2B 的网络地址和 2B 的主机地址组成，网络地址的最高位必须是 10，地址范围 128.1.0.1～191.255.255.254 可用的 B 类网络有 16 384 个，每个网络能容纳 65 534 个主机。

③ C 类 IP 地址。一个 C 类 IP 地址由 3B 的网络地址和 1B 的主机地址组成，网络地址的最高位必须是 110。地址范围 192.0.1.1～223.255.255.254 可用的 C 类网络可达 2 097 152 个，每个网络能容纳 254 个主机。

④ D 类 IP 地址。用于多点广播(Multicast)。D 类 IP 地址第一个字节以 1110 开始，它是一个专门保留的地址。地址范围是 224.0.0.1～239.255.255.254。

⑤ E 类 IP 地址。以 1111 开始，为将来使用保留。

另外，一个 C 类 IP 地址的 4 段号码中，前 3 段号码为网络号码，剩下的 1 段号码为本地计算机的号码。如果用二进制表示 IP 地址的话，C 类 IP 地址就由 3B(24 位)的网络地址和 1B 的主机地址组成。

在本题中，0/18 说明前 18 位表示网络号，所以子网个数占 24−18=6 位，子网数为 $2^6=64$ 个。另外，全零(0. 0. 0. 0)地址指任意网络。全 1 的 IP 地址(255. 255. 255. 255)是当前子网的广播地址。因此，该校园网包含 64−2=62 个 C 类网络。

(67)【答案】B

【解析】DHCP(Dynamic Host Configuration Protocol，动态主机分配协议)分为两个部分：一个是服务器端，另一个是客户端。所有的 IP 网络设定数据都由 DHCP 服务器集中管理，并负责处理客户端的 DHCP 要求；而客户端则会使用从服务器分配下来的 IP 环境数据。

在 DHCP 中，需要 IP 地址的主机用它的 MAC 地址广播一个 DHCP discover 分组。DHCP 服务器用一个 DHCP offer 分组进行应答，应答分组中包括没被使用的 IP，主机在得到的 IP 地址中选择一个，并用 DHCP request 分组广播它的选择，被选定的服务器用 DHCP ack 进行确认。分配出的 IP 地址有生命期，必须定期刷新以保持它的有效性。当主机完成任务以后，发送一个 DHCP release 分组释放占用的 IP 地址，否则当超过生命期后，地址自动被释放。

从 DHCP 的原理可以看出子网内是可以有多个 DHCP 的，用户机以收到的第一个 DHCP 应答信号为准，进行 IP 获取。显然 B 描述是正确的。

(68)【答案】D

【解析】POP3(Post Office Protocol 3)即邮局协议的第三个版本，它规定怎样将个人计算机连接到Internet 的邮件服务器和下载电子邮件的电子协议。它是因特网电子邮件的第一个离线协议标准，POP3 允许用户从服务器上把邮件存储到本地主机(即自己的计算机)上，同时删除保存在邮件服务器上的邮件，而POP3 服务器则是遵循 POP3 协议的接收邮件服务器，用来接收电子邮件。

(69)【答案】C

【解析】“＜title style= " italic " ＞science＜/title＞”是一个 XML 元素的定义，其中，title 是元素标记名称；style 是元素标记属性名称；italic 是元素标记属性值；science 是元素内容。

(70)【答案】A

【解析】本题考查多模光纤与单模光纤的区别。

多模光纤的特点是成本低，芯线宽，耗散大，低效，用于低速度、短距离的通信。

单模光纤的特点是成本高，芯线窄，需要激光源，耗散小，高效，用于高速度、长距离的通信。

(71)~(75)　【答案】(71)B；(72)D；(73)A；(74)C；(75)D

参考译文：统一软件开发过程是一种软件工程的处理方式，它记录了许多现代软件发展过程中很好的惯例。用例和场景的概念已经被证明是一个用于记录功能需求的好方法。RUP 能从两方面进行描述：时间和内容。在时间方面，软件生命周期被分解成许多个小周期。每个周期又被分成四个连续的状态，这些状态在定义好的界限点处结束并能进一步被分解成迭代块(迭代块是指最终使得一个可执行的产品能够生成一个发布的完整的循环块，处在不断演化中的最终产品的一个子集)，而这个迭代块将不断演化增加，最终成为一个完整的系统。内容结构是指一种方法，该方法是指自然地逻辑性地组织活动。

下午科目答案与解析

试题一

【答案】

【问题 1】：(1)费用单　(2)待租赁房屋列表　(3)看房请求　(4)变更房屋状态请求

【问题 2】：(5)房主信息文件 (6)租赁者信息文件 (7)房屋信息文件 (8)看房记录文件

【问题 3】：(1)起点：房主；终点：变更房屋状态；数据流名称：变更房屋状态请求

(2)起点：租赁者；终点：登记租赁者信息；数据流名称：租赁者信息

(3)起点：租赁者；终点：安排租赁者看房；数据流名称：看房请求

【解析】本题考查的是分层数据流图，该题型每年必考，是需要重点掌握的内容。

解题的两大原则：数据平衡原则，系统功能描述与数据流图的一致性原则。

首先根据数据平衡原则有：

在 0 层图中，与“房主”相关的数据流有五条。根据数据平衡原则顶层图应有与之对应的数据流，但“费用单”数据流在顶层图中找不到，所以(1)应是“费用单”数据流。

通过比较顶层图和 0 层图中与外部实体“租赁者”相关的数据流，可以发现：出现在 0 层图上的数据流“待租赁房屋列表”是顶层图上没有的，且与(2)处的数据流方向一致。由此可以判定，(2)处的数据流就是“待租赁房屋列表”。而顶层图中的数据流“租赁者信息”却是 0 层图上没有的。这样就找到了 0 层图上缺失的第二条数据流：租赁者信息，它的起点是“租赁者”，终点是加工“登记租赁者信息”。

根据系统功能描述与数据流图的一致性原则有：

由于(4)处缺失的数据流是一条输入数据流，从说明中可以看出，只有功能 6“当租赁者与房主达成租

房或退房协议后，房主向系统提交变更房屋状态的请求”所描述的数据流没有在“房主”与系统之间体现出来。因此可以确定，(4)处缺失的数据流就是“变更房屋状态请求”。相应地，可以确定，在 0 层图中缺失的其中一条数据流也是它，其起点是“房主”，终点是“变更房屋状态”这个加工。

由于说明中有“租赁者”相关的功能“一旦租赁者从中找到合适的房屋，就可以提出看房请求”，这一功能未在图中体现出来。这样就能确定(3)处的数据流应该是“看房请求”。而 0 层图中也没有出现这条数据流。所以，0 层图中缺失的第三条数据流就是“看房请求”，它的起点是“租赁者”，终点是加工“安排租赁者看房”。

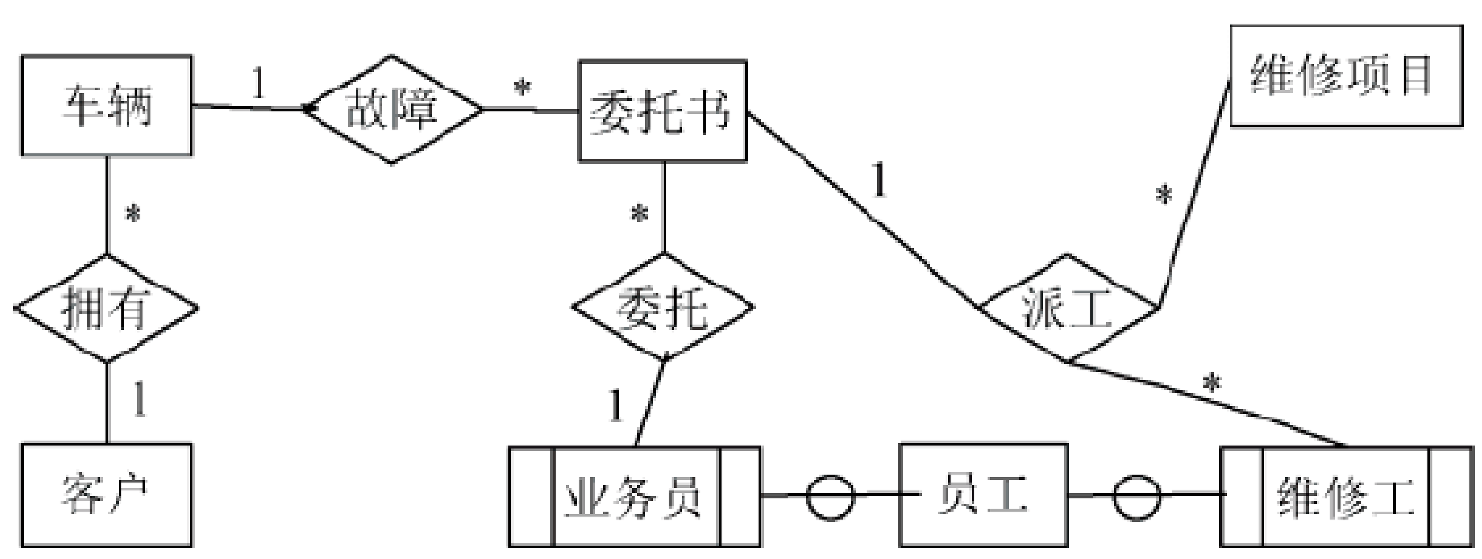
由说明的描述可以得知，本系统中的数据存储有：房主信息文件、房屋信息文件、租赁者信息文件、看房记录文件。下面就可以根据相应的加工对号入座了。显然，(5)处的是房主信息文件；(6)处的是租赁者信息文件；(7)处的是房屋信息文件；(8)处的是看房记录文件。

试题二

【答案】

【问题 1】(1) *n* 或 *m* 或*　(2) 1　(3) *n* 或 *m* 或*　(4) *n* 或 *m* 或*

【问题 2】完整的实体联系图如下图所示。



【问题 3】(5)　客户编号，客户名称，客户性质

(6) 委托书编号，客户编号，车牌号，业务员编号　或：委托书编号，车牌号，业务员编号

(7) 委托书编号，维修工编号，维修项目编号

(8) 员工编号，员工姓名

【问题 4】客户：客户编号

委托书：委托书编号

派工单：委托书编号，维修项目编号，维修工编号

【解析】本题考查数据库设计，设计考点有数据库的概念结构设计和逻辑结构设计。

【问题 1】由维修委托书的故障描述，维修类型、作业分类，可知，一台车可能有多个故障，对应多个维修委托书，所以(1)空填写*；由题目中“维修车间根据维修委托书和车辆的故障现象，在已有的维修项目中选择并确定一个或多个具体维修项目，安排相关的维修工及工时，生成维修派工单”可知，一份委托书包含了一个或多个维修项目，而每个维修项目可以由多个维修工来完成，每一个维修工又可以完成多个维修项目，所以(2)空填写 1，(3)、(4)填写*。

【问题 2】需要补充车辆和客户之间以及委托书和业务员之间的关系。由题目“一个客户至少拥有一台车”可知，客户和车辆之间是“拥有”关系，且是一对多的关系；在由题目中“业务员对车辆进行检查和故障分析后，与客户磋商，确定故障现象，生成维修委托书”可知，业务员与委托书之间是“委托”关系，且一名业务员可以受理多份委托书，而一份委托书由一名业务员来生成。

【问题 3】本题又是补充逻辑结构设计题，几乎每年都考，这类题目只要仔细看需求分析结果或者仔

细观察题目中已知的表，很容易就能做出，关键是需要细心，不要漏掉什么属性。根据客户和车辆信息表可知，客户关系应包括客户编号、客户名称、客户性质、折扣率、联系人等属性，主键显然为客户编号；而车辆关系应包括车牌号、客户编号、车型、颜色、车辆类别等属性，主键为车牌号。根据维修委托书表可知委托书应包括委托书编号、车牌号、客户编号、业务员编号、维修类型等属性，其主键为委托书编号。根据维修派工单可知，派工单应包括委托书编号、维修项目编号、维修工编号、工时等属性，主键是委托书编号、维修项目编号和维修员编号。根据实体联系图知，员工包括业务员和维修工，他们共有的属性是员工编号、员工姓名、工种、员工类型、级别等属性，主键为员工编号。

【问题 4】参考问题 3 的分析。

试题三

【答案】

【问题 1】(a)资源目录　(b)图书　(c)唱片

【问题 2】CatalogItem 的属性：索引号、名称、出版时间、资源状态

图书的属性：作者、页数

唱片的属性：演唱者、介质类型

【问题 3】(1)1　(2)0..*　(3)1　(4)0..*　(5)1　(6)1 或者 0..1

【解析】本题主要考查 UML 中的类图设计，题目的三个问题都是对类图的元素进行补充。类图的设计是根据系统的功能需求而来的，所以解题的关键在于对“系统功能说明”的理解。下面将通过对“系统功能说明”的分析，来解答试题。

从系统功能说明中的“图书管理系统的资源目录中记录着所有可供读者阅读的资源”和“资源可分为两类：图书和唱片”，可以得知 1 个资源目录中对应着多个可供读者借阅的资源，这些资源分为图书类与唱片类，所以(a)为资源目录，(b)和(c)分别为图书和唱片，同时(1)应填 1，(2)应填 0..*。(所有的可供读者借阅资源数有可能为 0，即还未录入任何资源的状态)

从“每项资源都有一个唯一的索引号。系统需登记每项资源的名称、出版时间和资源状态”可以得知，资源目录中的每项资源，即类图中的 CatalogItem，有索引号、名称、出版时间和资源状态这四个关键属性。

从“对于图书，系统还需登记作者和页数；对于唱片，还需登记演唱者和介质类型(CD 或者磁带)”，可以得知图书有作者和页数两个关键属性，唱片有演唱者和介质类型两个关键属性。

Borrower 代表读者，而 BorrowerItems 为借书记录文件，同时系统功能说明中有“系统为每个读者创建了一个借书记录文件，用来保存读者所借资源的相关信息”，所以它们之间的关系应为 1 对 1，即第(5)空和第(6)空均填 1。

试题四

【答案】(1)EnQueue(&tempQ,root)　(2)brotherptr=brotherptr->nextbrother

(3)!IsEmpty(tempQ)　(4)DeQueue(&tempQ,&ptr)　(5)!ptr->firstchild

(6)EnQueue(&tempQ,ptr->firstchild)　(7)brotherptr=brotherptr->nextbrother

【解析】解答此题的关键在于理解用队列层序遍历树的过程。算法的流程是这样的：首先将树根结点入队，然后将其所有兄弟结点入队(当然，由于是根结点，故无兄弟结点)；完成这一操作以后，便开始出队、打印；在打印完后，需要进行一个判断，判断当前结点有无孩子结点，若有孩子结点，则将孩子结点入队，同时将孩子结点的所有兄弟结点入队；然后继续进行出队操作，出队后再次判断当前结点是否有孩

子结点，并重复上述过程，直至所有结点输出。

接下来以本题为例来说明此过程。首先将树根结点 D 入队，并同时检查是否有兄弟结点，对于兄弟结点则一并入队。这里的 D 没有兄弟结点，所以队列此时应是 D。

接下来执行出队操作。D 出队，出队以后检查 D 是否有子结点，经检查，D 有子结点 B，所以将 B 入队，同时将 B 的兄弟结点 A 和 E 按顺序入队。得到队列 B、A、E。

接下来再执行出队操作。B 出队，同时检查 B 是否有子结点，B 无子结点，所以继续执行出队操作。A 出队，同时检查 A 是否有子结点，A 有子结点 F，所以将 F 入队，同时将 F 的兄弟结点 P 入队。得到队列 E、F、P。

接下来再次执行出队操作。E 出队，E 有子结点 C，所以 C 出队。得队列 F、P、C。

接下来再次执行出队操作。F 出队，F 无子结点，继续出队操作，P 出队，P 仍无子结点，最后 C 出队，整个过程结束。

通过对算法的详细分析，我们可以轻松得到答案。(1)应是对根结点 root 执行入队操作，即 EnQueue(&tempQ,root)。(2)在一个循环当中，循环变量是 brotherptr，此变量无语句对其进行更新，所以(2)必定是更新 brotherptr。结合前面的算法分析可知(2)应填 brotherptr=brotherptr->nextbrother。(3)、(4)加上后面的语句 "printf("%c\t", ptr->data)；"是控制数据的输出，这些数据应是从队列中得到，所以此处必有出队操作，同时在出队之前应判断队列是否为空，所以(3)、(4)填!IsEmpty(tempQ)和 DeQueue(&tempQ,&ptr)。(5)实际上是问"在什么情况下，要持续进行出队操作？"，前面的算法分析中已指出：若出队结点无子结点，则继续进行出队操作，所以(5)填!ptr->firstchild。(6)和(7)所在的语句段的功能是将刚出队结点的子结点及其兄弟结点入队，所以(6)填 EnQueue(&tempQ,ptr->firstchild)。(7)和(2)相同，填 brotherptr=brotherptr->nextbrother。

试题五

【答案】(1)std　(2)virtual void　(3)virtual void

(4)flyBehavior　(5)quackBehavior　(6)flyBehavior->fly()

(7)quackBehavior->quack()　(8)FlyNoWay()　(9)Squeak()

【解析】本题考查面向对象的程序设计的抽象类和多态。程序解释如下：

(1) using namespace (1) std;　/*所谓 namespace，是指标识符的各种可见范围。C++标准程序库中的所有标识符都被定义于一个名为 std 的 namespace 中。*/

(2) 由题中信息：类 FlyBehavior 与 QuackBehavior 为抽象类，分别用于表示抽象的飞行行为与发声行为，而这两种行为对应的不止是一种方式，为了表示多种行为方式(即多态)，需要定义虚函数，用以给子类进行重载。再由子类中的函数 void fly( )及 void quack( )可知，父类中的对应函数返回值也应为 void。所以(2)和(3)填空如下：

```
class FlyBehavior{
public:  (2) virtual void fly( )=0;
};
class QuackBehavior{
public:  (3) virtual void quack( )=0;
}
```

(3) Duck 类的定义，由子类 RubberDuck 类中出现的 flyBehavior=new lyNoWay();quackBehavior=new Squeak();可知这两个指针类型对象为 flyBehavior 和 quackBehavior。所以(4)和(5)填空如下：

```
Class Duck{
    protected:
    FlyBehavior * _(4) flyBehavior_;
    QuackBehavior * _(5) quackBehavior_;
```

由 FlyBehavior * flyBehavior; QuackBehavior * quackBehavior;及函数名可知，该函数体应调用 FlyBehavior 类中的 fly 函数和 QuackBehavior 类中的 quack 函数。所以(6)和(7)填空如下：

```
public:
    void fly(){  _(6) flyBehavior->fly ()_ ;  }
    void quack() {  _(7) quackBehavior->quack ()_ ; };
    virtual void display()=0;
};
```

(4) 由题中所给信息可知 RubberDuck 是“发出空气与橡皮摩擦声”并且“不能飞行”的鸭子种类，因此构造函数中创建的对象 flyBehavior 和 quackBehavior 对应的类型分别为 FlyNoWay、Squeak，所以(8)和(9)填空如下：

```
class RubberDuck: public Duck {
public:
    RubberDuck( ){
    flyBehavior=new (8) FlyNoWay () ;
    quackBehavior=new (9) Squeak () ;
}
```

试题六

【答案】(1)interface　(2)interface　(3)flyBehavior
(4)quackBehavior　(5)flyBehavior.fly　(6)quackBehavior.quack()
(7)abstract　(8)FlyNoWay()　(9)Squeak()

【解析】本题考查面向对象的程序设计的抽象类和多态，使用的语言是 Java。

(1) 由题中所给信息：接口 FlyBehavior 与 QuackBehavior 分别用于表示抽象的飞行行为与发声行为，可知(1)处和(2)处都应填 interface。

(2) 由子类 RubberDuck 类中出现的 flyBehavior 和 quackBehavior，而它们在之前并没有被定义过，可知(3)处和(4)处需声明这两个对象分别为 FlyBehavior、QuackBehavior 类型，否则出错。

(3) 由 FlyBehavior flyBehavior、QuackBehavior quackBehavior 及函数名可知该函数体应调用 FlyBehavior 类中的 fly 函数和 QuackBehavior 类中的 quack 函数。所以(5)处和(6)处应分别填写 flyBehavior.fly 和 quackBehavior.quack()。

(4) 由于 Duck 类为抽象类，因此需要一个抽象函数来描述鸭子的总体外观特征，所以(7)处应填写 abstract。

对于题目第(8)处和第(9)处，由题中所给信息可知 RubberDuck 是“发出空气与橡皮摩擦声”并且“不能飞行”的鸭子种类，因此构造函数中创建的对象 flyBehavior 和 quackBehavior 对应的类型分别为 FlyNoWay、Squeak。

参 考 文 献

[1] 胡圣明. 软件设计师教程(第 3 版)(修订版). 北京：清华大学出版社，2011
[2] 全国计算机专业技术资格考试办公室. 软件设计师历年试题分析与解答(2009—2010). 北京：清华大学出版社，2011
[3] 何光明，金静文. 软件设计师考试同步辅导(上午科目)(第 2 版). 北京：清华大学出版社，2010
[4] 张宏，王宏华. 软件设计师考试同步辅导(下午科目)(第 2 版). 北京：清华大学出版社，2010
[5] 胡圣明，张淑平，霍秋艳. 软件设计师教程. 北京：高等教育出版社，2010
[6] 全国计算机专业技术资格考试办公室. 软件设计师考试大纲与培训指南. 北京：清华大学出版社，2009
[7] 雷震甲. 网络工程师教程(第 3 版)(修订版). 北京：清华大学出版社，2011